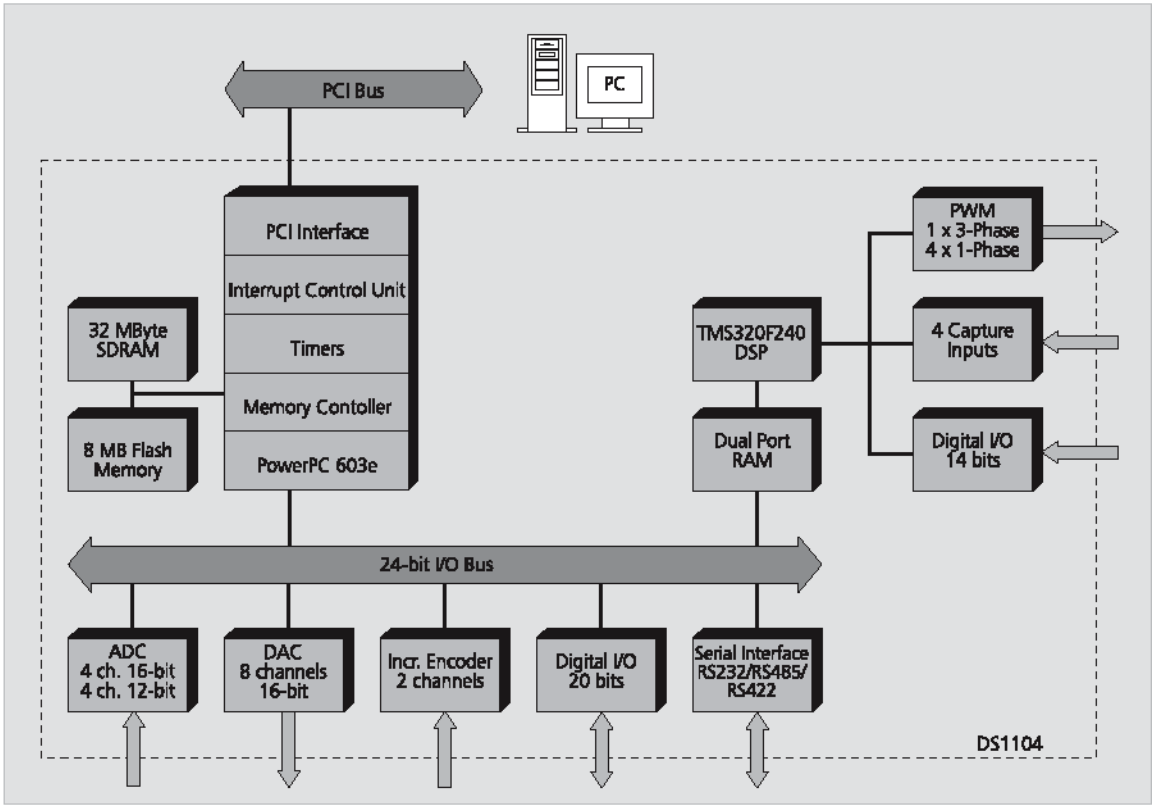
Laboratoire GREYC UMR 6072

Equipe Automatique

Plate-forme PS600 et carte DSP DS1104



Dossier réalisé par

Philippe DORLEANS & J.F. MASSIEU

Table des Matières

I - Introduction 1
II - Présentation du système 2
III - Principe général de fonctionnement du système 4
III.1 - Bloc 'Manip' du schéma général Simulink 7
III.2 - Bloc 'Modèle' du schéma général Simulink 8
III.2.1 - Construction du modèle non-linéaire sous Simulink 11
III.2.2 - Linéarisation des équations et fonction de transfert 12
III.2.2.1 - Pendule en position basse ($\theta = 0$) 13
III.2.2.2 - Pendule en position haute ($\theta = \pi$) 14
III.2.3 - Représentation dans l'espace d'état 16
III.2.4 - Identification de certains paramètres physiques du système 17
III.3 - Bloc 'Standup Routine' 19
III.4 - Bloc 'Commande' 20
ANNEXES 24
ANNEXE I : Modes de pilotage 25
Pilotage manuel (repère ①) 25
Pilotage externe (repère ②) 25
Pilotage automatique 25
ANNEXE II : Carte dSPACE 1104 27
ANNEXE III : Control Desk 29
Démarrer un nouveau projet sous Control Desk 29
Ouvrir un projet existant 30
Travailler dans l'espace de travail 30
- Créer un layout 30
- Sauvegarder des données sur fichier 31

La fenêtre d'outils 33
ANNEXE IV : Modes de marche 34
ANNEXE V : Standup routine 35
ANNEXE VI : Schéma de câblage 40

I - Introduction

L'équipe automatique du Greyc dispose d'un ensemble de procédés ayant des caractéristiques spécifiques (systèmes à retard, systèmes flexibles, systèmes instables…) et permettant de valider des méthodologies développées sur site. Ce document concerne la plate-forme PS 600 de chez Amira, système qui a la particularité d'être modifié par l'intermédiaire de différents modules fixés sur un support mobile. C'est ainsi que la PS600 peut être configurée en pendule inverse (simple ou double) ou en pont roulant. Le document proposé concerne le système du pendule inverse simple, système très intéressant principalement pour ses propriétés de non-linéarité et d'instabilité.

L'objectif est de développer et d'implémenter une commande numérique en temps réel de manière à maintenir le pendule en position verticale haute. Il s'agit ici de répondre à un problème de régulation. Ce document a été rédigé sous la forme d'un manuel d'utilisation et récapitule les travaux déjà réalisés. Ceci doit permettre d'effectuer une démarche complète pour la synthèse d'un correcteur. L'analyse instrumentale du système étant effectuée, il n'y a plus que l'implémentation de la loi de commande à réaliser. Le système est interfacé avec un PC et la communication entre le système et le PC se fait via une carte d'acquisition de type DSP (carte dSPACE 1104) et en utilisant les environnements logiciels Matlab/Simulink et Control Desk.

La première partie du document est consacrée à la description précise du système et de son interfaçage avec le PC. Ensuite est traité en détail l'environnement logiciel c'est-à-dire le schéma général sous Simulink associé à une interface graphique développée sous Control Desk. Dans cet environnement apparaissent des blocs distincts faisant l'objet, pour chacun d'eux, d'un développement complet. Le premier bloc traité concerne la modélisation physique du système qui est établie en utilisant les équations du mouvement par la méthode de Lagrange suivie d'une linéarisation autour des deux points d'équilibre possibles. Est traitée également dans cette partie l'implémentation sous Simulink du modèle non-linéaire. La linéarisation autour des deux points d'équilibre a conduit à la construction du modèle d'état nécessaire pour la synthèse d'une loi de commande. Dans ce document est présentée une loi de commande par retour d'état de type LQ.

La structure du schéma général développé à partir de l'environnement Simulink a été conçue de telle manière à être complètement adaptable à l'intégration d'un nouveau correcteur en modifiant uniquement le bloc 'Commande' du schéma général.

II - Présentation du système

Le système est constitué d'un chariot mobile se déplaçant sur un rail linéaire de 120 cm et portant un balancier pouvant pivoter autour d'un axe fixé au chariot. Un moteur à courant continu (actionneur) est relié au chariot par une courroie (cf. figures II.1 et II.2). Sous l'application d'une tension de commande, le moteur exerce une force longitudinale F(t) ce qui entraîne le déplacement du chariot. Deux capteurs de fin de course sont disposés de manière à détecter les extrémités du rail. L'objectif final est de contrôler le déplacement du chariot de manière à maintenir le pendule en position d'équilibre vertical vers le haut.

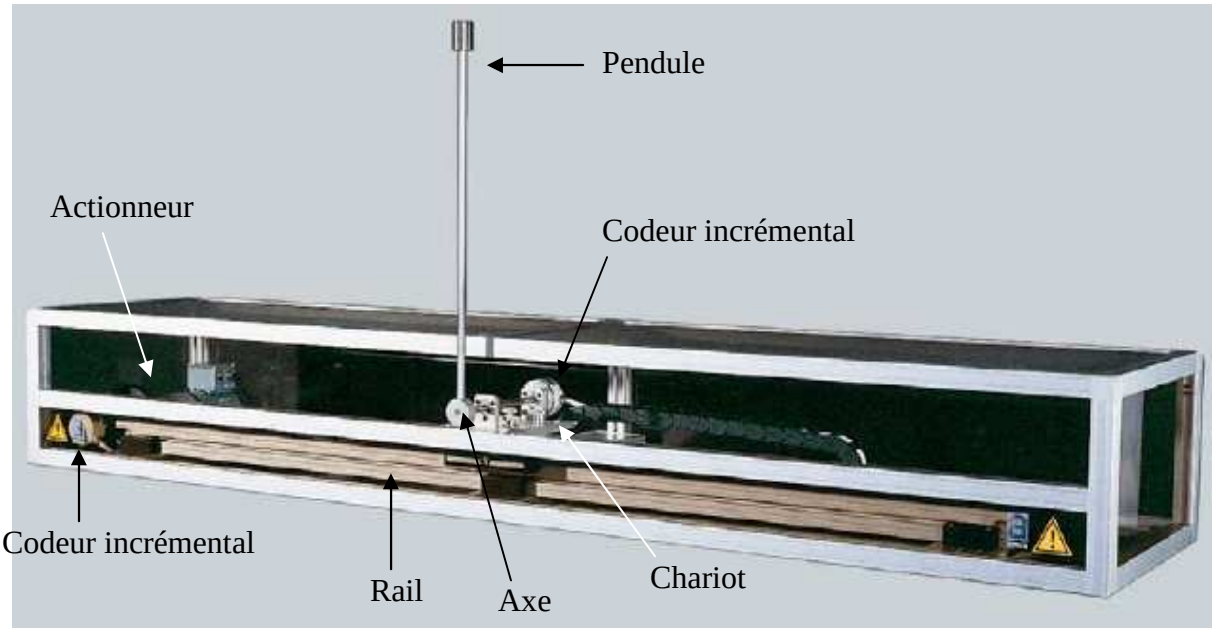


Figure II.1 : Le pendule inverse PS 600 de chez Amira

Les capteurs sont constitués de deux codeurs incrémentaux (type RI 58D), l'un fournissant la positon z(t) du chariot et sa vitesse linéaire $\dot{z}(t)$ et l'autre fournissant la position angulaire du pendule $\theta(t)$ et sa vitesse angulaire $\dot{\theta}$.

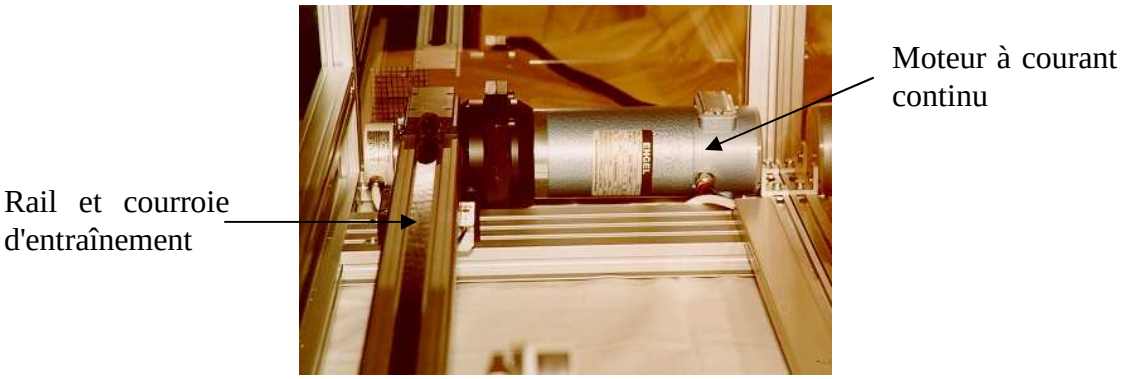


Figure II.2 : L'actionneur

Pour effectuer des asservissements numériques, le système est équipé d'un bloc électronique de commande et communique avec un ordinateur via une carte d'acquisition DSP 1104 de

chez Dspace (cf. figures II.3 et II.4). L'ensemble fonctionne sous les environnements logiciels Matlab/Simulink et Control Desk.

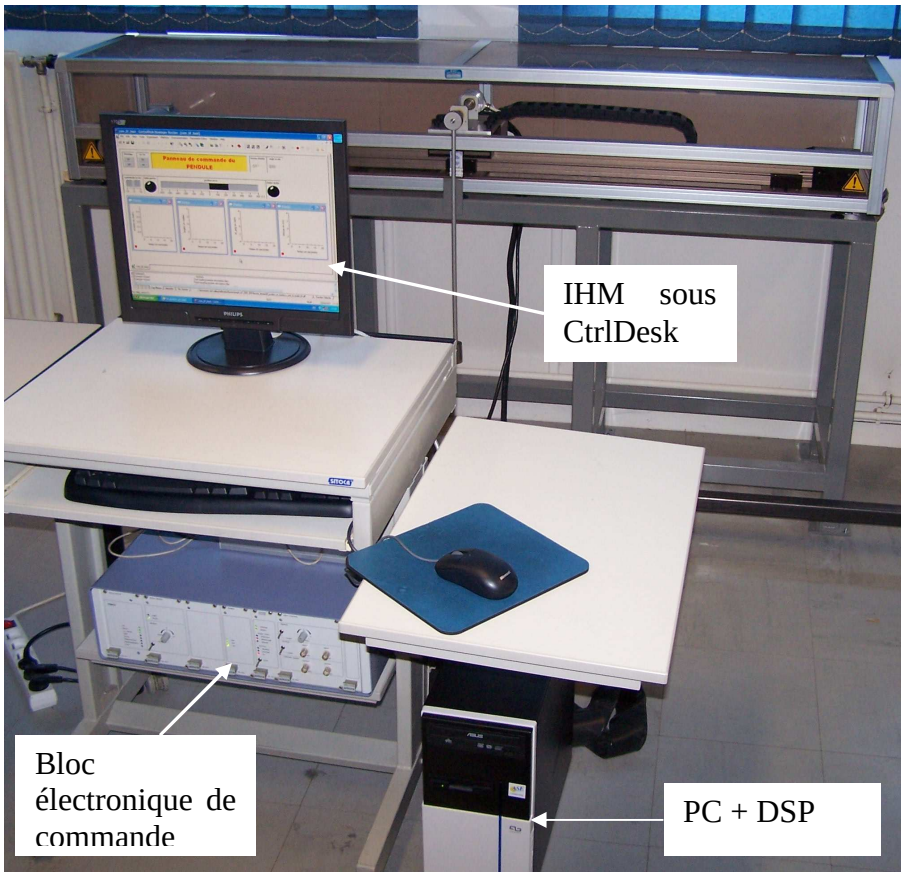


Figure II.3 : Système complet

Les signaux du système sont récupérés en face arrière du bloc de commande par l'intermédiaire d'un connecteur 50 broches et sont connectés à la carte DSP. Le plan de câblage entre le bloc de commande et la carte DSP est donné en annexe VI. On récupère ainsi à partir de ces signaux la position et la vitesse linéaire du chariot mais aussi la position angulaire et la vitesse angulaire du pendule. Le système global peut être schématisé de la manière suivante :

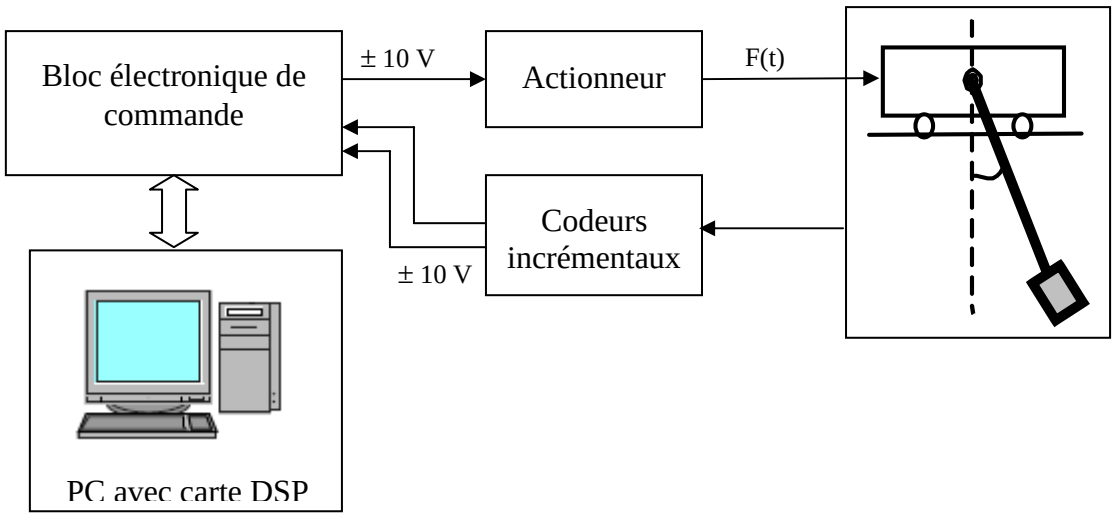


Figure II.4 : Schéma du système complet

La face avant du bloc de commande dispose d'interrupteurs pour la sélection des modes de fonctionnement, de potentiomètres pour le réglage en mode manuel de la tension de commande et de LEDS pour la visualisation de l'état de la plate-forme. On y trouve également des connecteurs BNC permettant de fournir une commande en tension externe ou pour relever des informations comme la position du chariot.

Des signaux d'informations sont également récupérés à l'aide des entrées numériques de la carte tels que les butées gauche et droite du chariot, le mode PC, etc.…

La sortie du DAC 4 (Digital Analogic Converter) permet d'envoyer au bloc de commande les signaux de commande analogiques pour déplacer le chariot. Les niveaux de tension analogique en entrée ou en sortie sont compris entre –10V et +10V.

III - Principe général de fonctionnement du système

Le principe général de mode de fonctionnement du système est illustré par le schéma Simulink donné ci-après.

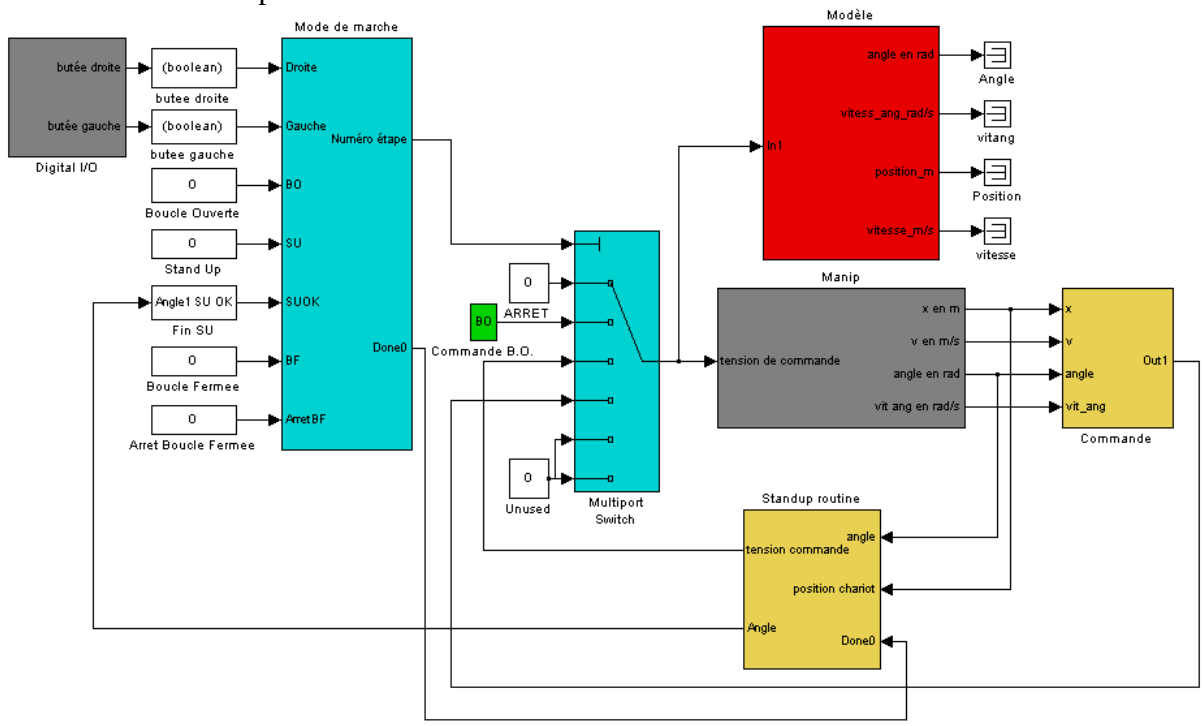


Figure III.1 : Schéma général Simulink de commande du pendule inverse

On distingue différentes parties ou blocs sur ce schéma Simulink :

- ✓ Un bloc "**Modes de Marche**" permet le choix, à partir de l'interface Controldesk, entre le fonctionnement en "boucle ouverte", le fonctionnement en "boucle fermée" et la procédure d'entraînement "Standup Routine".

- ✓ Un bloc "**Modèle**" dans lequel est établi un modèle non-linéaire et un modèle linéarisé du système.

- ✓ Un bloc "**Manip**" qui rend compte simplement des échanges avec le système réel (entrées/sorties).

- ✓ Un bloc "**Commande**" : c'est dans ce bloc que doit être implanté l'algorithme de commande pour le maintien du pendule en position verticale.

- ✓ Un bloc "**Standup Routine**" qui permet d'amener automatiquement le pendule dans une position proche de l'équilibre vertical.

- ✓ Un bloc "**Digital I/O**" qui permet d'effectuer un RAZ des codeurs incrémentaux, de générer les signaux de déblocage pour le bloc électronique de commande (cf. annexe I) et de récupérer les informations des butées droite et gauche pour la détection des extrémités du rail.

Dans ce rapport, chacun des blocs de ce schéma fait l'objet d'un développement précis.

A ce schéma Simulink a été développé un environnement graphique présenté figure III.2 permettant d'une part, le dialogue en temps réel avec les entrées – sorties analogiques et numériques et d'autre part, la possibilité de modifier des paramètres internes aux instruments graphiques. La méthode pour concevoir une interface graphique dans l'environnement Control Desk est entièrement développée l'annexe III.

L'interface graphique développée pour le système du pendule est présentée figure suivante.

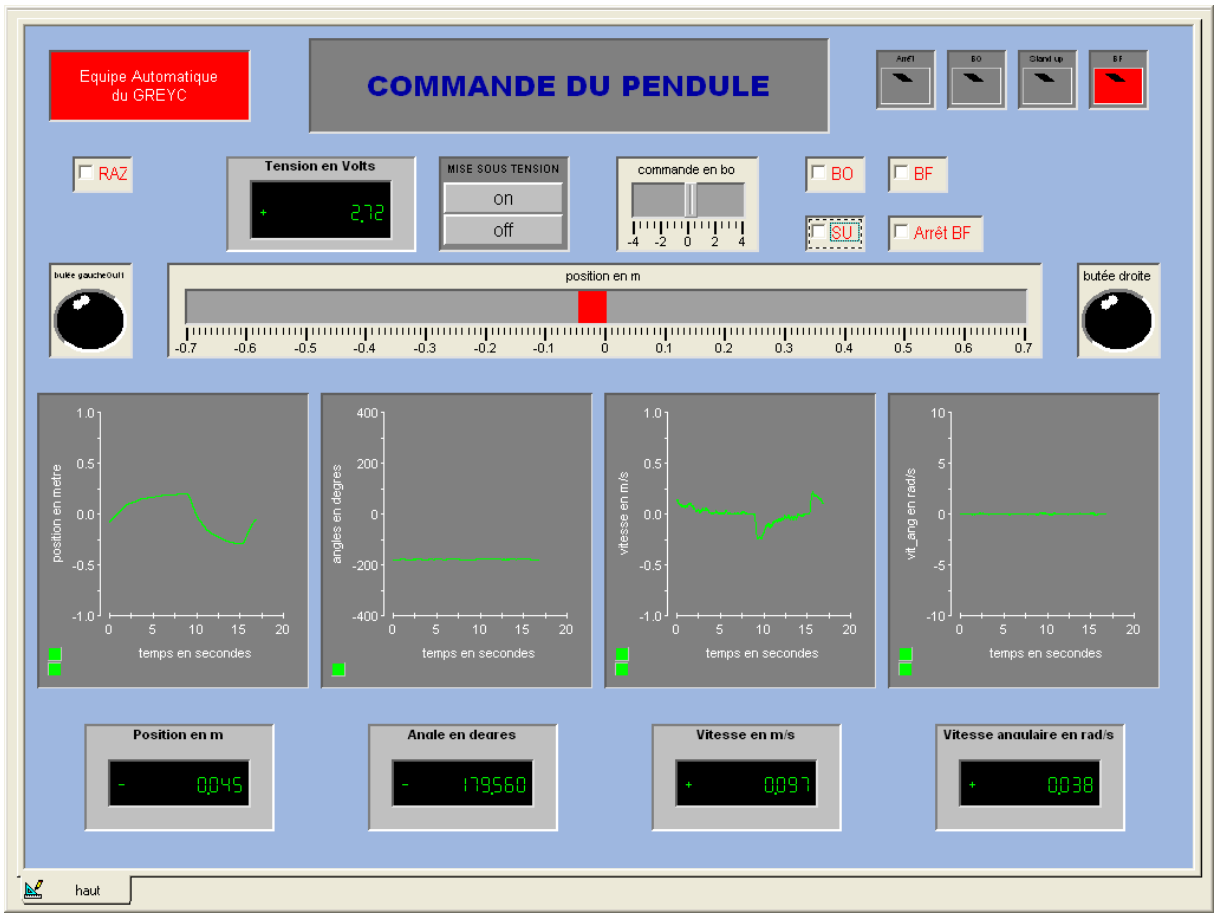


Figure III.2 : Interface graphique développée sous Control Desk

L'animation de cette interface graphique est réalisée en cliquant sur l'icône de l'environnement Control Desk et ensuite, il faut cliquer sur **'on'** du bouton '**MISE SOUS TENSION**'.

A chaque lancement du programme chargé dans le DSP, le système est automatiquement initialisé en mode "boucle ouverte" c'est-à-dire que le chariot ne peut être déplacé qu'à partir de commandes manuelles en actionnant le curseur **'Commande en bo'** de l'interface Control Desk. Le déplacement du curseur entraîne une variation de la tension analogique appliquée au moteur comprise entre – 4V et + 4V. Ceci permet d'amener le chariot à la position centrale afin de faire une prise de référence. La prise de référence est réalisée en cliquant sur l'objet **'RAZ'** ce qui a pour conséquence d'initialiser les codeurs incrémentaux à zéro.

En effet, la position du chariot et l'angle du pendule sont donnés par rapport à la position et à l'inclinaison lors du chargement du programme dans la carte DSP. C'est pourquoi, au début de la manipulation, on doit placer le chariot à la position repérée comme étant le zéro (position de référence) située au milieu du banc. Ceci est dû au fait que les codeurs incrémentaux fournissent une information relative. Il est donc nécessaire d'effectuer un RAZ avant toute manipulation.

A partir du mode "**boucle ouverte**", on peut passer soit en mode "**boucle fermée**", soit en mode "**Standup Routine**".

- Si on décide de passer directement en mode "**boucle fermée**", il faut prendre soin de placer manuellement le pendule dans une position voisine de l'équilibre vertical puis lancer le mode "boucle fermée".

- Si on décide de passer en mode "**Standup Routine**" (cliquer sur '**SU**'), le pendule va être amené automatiquement dans une position proche de l'équilibre vertical et dès que le voisinage immédiat de cette position est atteint, le système bascule automatiquement en mode "boucle fermée". Le système reste alors dans ce mode de marche tant que l'arrêt n'est pas demandé (manuellement à partir de l'interface graphique de Control Desk) ou qu'aucune butée (gauche ou droite) n'est atteinte. Ces spécifications fonctionnelles sont reportées sur un diagramme grafcet fourni en annexe V et sont gérées sous Simulink par le bloc 'Mode de marche'.

Quatre graphiques permettent de visualiser en fonction du temps (de la gauche vers la droite) la position du chariot, la position angulaire du pendule, la vitesse du chariot et la vitesse angulaire du pendule. A chaque graphe est associé un indicateur numérique permettant d'afficher la valeur exacte de la grandeur mesurée.

Un curseur central permet également de visualiser la position du chariot auquel est associé 2 voyants indiquant si le chariot a atteint une des extrémités du rail. Si c'est le cas, le voyant concerné s'allume en rouge et un dispositif de sécurité fait passer la tension de commande à zéro. Cette tension est également affichée au moyen d'un indicateur numérique.

Si une extrémité est atteinte, il faut recommencer la procédure de mise en marche c'est-à-dire passer en mode "boucle ouverte", déplacer le chariot puis RAZ, etc.…

III.1 - Bloc 'Manip' du schéma général Simulink

Le contenu du bloc '**Manip**' est présenté figure suivante. On y trouve principalement des convertisseurs DAC et ADC ainsi que des coefficients de mise à l'échelle.

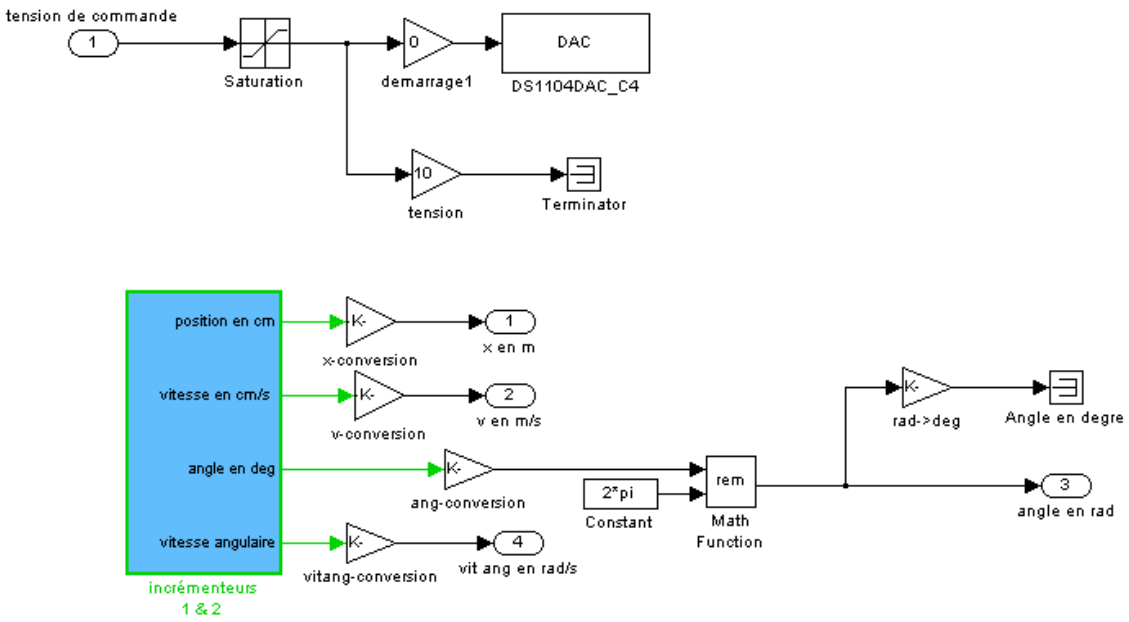


Figure III.3 : Contenu du bloc 'Manip'

Bien que la carte DSP permette de fonctionner à une cadence très rapide (quartz à 250 MHz), pour notre application, la période d'échantillonnage a été fixée à 0,01 sec. Cette limitation est liée à des considérations matérielles et principalement aux codeurs incrémentaux. En effet, une fréquence plus élevée entraîne une mauvaise acquisition des vitesses linéaire et angulaire.

La tension de commande est appliquée à l'actionneur par l'intermédiaire du DAC 4 . Elle est au préalable limitée via un bloc '**Saturation**' de manière à ne pas dépasser ± 10 Volts. Il faut préciser qu'il existe un facteur 10 entre la valeur des signaux calculés ou mesurés sous Simulink et les signaux analogiques réels. Concrètement, cela signifie qu'un signal de valeur 0,5 par exemple envoyé au DAC engendre la génération d'un signal analogique de 5 V. Cette règle est également vraie pour l'acquisition. Par conséquent, les paramètres du bloc '**Saturation**' sont fixés à ± 1 ce qui entraîne une limitation en tension à ± 10 Volts.

Le gain '**tension**' a été introduit uniquement pour afficher à l'écran graphique la valeur de la tension de commande appliquée à l'actionneur. Le gain '**demarrage1**' est relié à l'objet graphique '**Mise sous tension**'. Il permet de contrôler le démarrage du système. En effet, un click de souris sur '**ON**' ou sur '**OFF**' conduit à lui affecter respectivement la valeur 1 ou la valeur 0. Cette dernière valeur entraîne une tension nulle et par conséquent, le chariot ne démarre pas.

Différents gains de conversion ont été introduits de manière à convertir les informations issues des capteurs en grandeurs physiques exprimées en unité SI. Dans le schéma Simulink, on trouve donc les conversions suivantes :

- Le gain **'x-conversion'** a été déterminé expérimentalement de la manière suivante : le codeur incrémental fournit la valeur de 7401,75 pour un déplacement du chariot de 1,2 m soit une conversion de 7401,75/1,2 c'est-à-dire 6168,12 pour un mètre. Le gain **'x-conversion'** a pour valeur : 1/6168,12.

- L'information de la vitesse linéaire du chariot se déduit du gain **'x-conversion'** en intégrant dans le calcul le facteur temps. En divisant **'x-conversion'** par la période d'échantillonnage (0,01 s), on obtient la valeur du gain **'v-conversion'**.

- Le facteur de conversion de l'angle du pendule est donné par le constructeur et vaut 4/45,5. Pour obtenir l'angle du pendule en radian, le gain **'ang-conversion'** a pour valeur : 4π / (45,5*180).

- La vitesse angulaire est obtenue en intégrant la période d'échantillonnage et donc la valeur du gain **'vitang-conversion'** est 4π / (45,5*180*0.01).

III.2 - Bloc 'Modèle' du schéma général Simulink

Cette partie concerne la modélisation physique du système afin d'établir un modèle dynamique de connaissance. Ce modèle est obtenu en établissant les équations du mouvement du pendule par la méthode de Lagrange. L'avantage de la méthode de Lagrange est sa généralité : elle est basée sur le principe de la conservation d'énergie mécanique. Le Lagrangien est défini comme la différence entre l'énergie cinétique et l'énergie potentielle du système. Par cette méthode, on obtient facilement les équations différentielles du mouvement du système.

Dans le modèle de connaissance, tous les paramètres physiques y apparaissent alors explicitement. Cette étape de modélisation permettra ensuite d'une part, l'obtention d'un modèle linéarisé autour d'un point de fonctionnement et donc la possibilité de synthétiser des lois de commande linéaires autour de ce point, et d'autre part, de valider les lois de commande développées à partir de la conception d'un simulateur du système.

L'ensemble chariot-pendule peut être représenté par le schéma suivant :

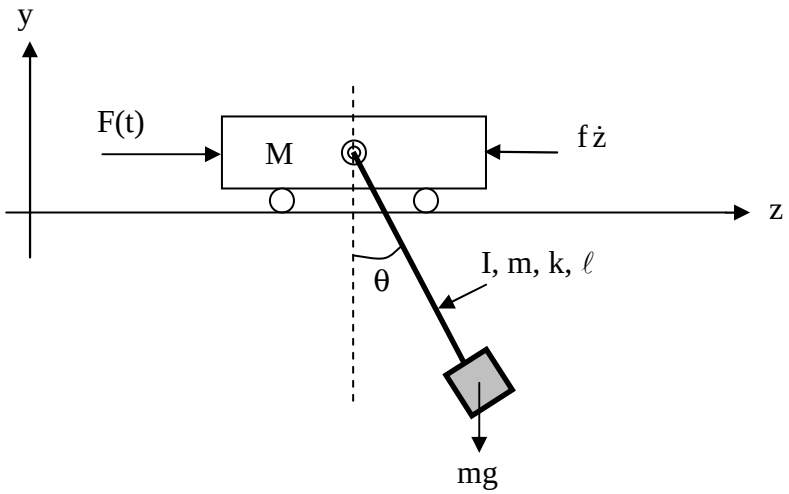


Figure III.4 : Schéma de l'ensemble chariot-pendule

Sur ce schéma apparaissent les différentes grandeurs physiques suivantes :

z(t)	Position et direction du chariot (en m)
θ(t)	Position angulaire du pendule (en rad)
F(t)	Force d'entraînement transmise par la courroie et due au moteur (en N)
M	Masse du chariot = 1,6 kg
m	Masse du pendule = 0,31 kg
g	Accélération de pesanteur = 9,81 m s^{-2}
ℓ	Longueur du pendule = 0,5 m
I	Moment d'inertie du pendule par rapport au centre de masse (en N m^2)
f	Coefficient de frottement visqueux entre le chariot et le rail et s'opposant à l'action F(t) (f = 210 N/m/s)
k	Coefficient de frottement visqueux du pendule sur son axe = 0.003 N/rad/s

Par la suite, on adoptera les notations suivantes :

$$\dot{\theta}=\frac{d\theta}{dt},\ \ddot{\theta}=\frac{d^2\theta}{dt^2},\ \dot{z}=\frac{dz}{dt}\ \text{ et }\ \ddot{z}=\frac{d^2z}{dt^2} \tag{3.1}$$

Pour une position z du chariot, la position du centre de masse du pendule en projetant sur les axes y et z est donnée par :

$$\vec{P}:\begin{bmatrix} z-\ell\sin\theta \\ -\ell\cos\theta \end{bmatrix} \tag{3.2}$$

La vitesse correspondante est donnée par : $\vec{V}:\frac{d}{dt}\begin{bmatrix} z-\ell\sin\theta \\ -\ell\cos\theta \end{bmatrix}=\begin{bmatrix} \dot{z}-\ell\dot{\theta}\cos\theta \\ \ell\dot{\theta}\sin\theta \end{bmatrix}$ (3.3)

Energie cinétique du chariot : $T_M=\frac{1}{2}M\dot{z}^2$ (3.4)

Energie cinétique du pendule : $T_m=\frac{1}{2}mV^2+\frac{1}{2}I\omega^2$ (3.5)

où I et ω représentent respectivement le moment d'inertie et la vitesse angulaire du centre de masse du pendule : $\omega=\dot{\theta}$. En utilisant les équations (3.3) et (3.5), on a :

$$T_m=\frac{1}{2}m\left(\dot{z}^2-2\dot{z}\ell\dot{\theta}\cos\theta+\ell^2\dot{\theta}^2\cos^2\theta+\ell^2\dot{\theta}^2\sin^2\theta\right)+\frac{1}{2}I\dot{\theta}^2 \tag{3.6}$$

L'énergie cinétique totale a pour expression :

$$T = T_M + T_m = \frac{1}{2} M \dot{z}^2 + \frac{1}{2} m \left(\dot{z}^2 - 2 \dot{z} \ell \dot{\theta} \cos\theta + \ell^2 \dot{\theta}^2 \cos^2\theta + \ell^2 \dot{\theta}^2 \sin^2\theta \right) + \frac{1}{2} I \dot{\theta}^2 \qquad (3.7)$$

Lorsque le chariot est en mouvement, l'énergie potentielle du système, fonction de l'angle du pendule, est donnée par la relation suivante :

$$V = -mg\ell\left(1 - \cos\theta\right) \qquad (3.8)$$

Le Lagrangien, noté L, est défini comme la différence entre l'énergie cinétique T et l'énergie potentielle V du système, L = T −V, par conséquent, en utilisant (3.7) et (3.8), on a :

$$L = \frac{1}{2} M \dot{z}^2 + \frac{1}{2} m \left(\dot{z}^2 - 2 \dot{z} \ell \dot{\theta} \cos\theta + \ell^2 \dot{\theta}^2 \cos^2\theta + \ell^2 \dot{\theta}^2 \sin^2\theta \right) + \frac{1}{2} I \dot{\theta}^2 + mg\ell\left(1 - \cos\theta\right) \qquad (3.9)$$

Les équations non-linéaires du mouvement vont être déterminées à partir de l'équation de Lagrange suivante :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\xi}}\right) - \frac{\partial L}{\partial \xi} = Q_\xi \qquad (3.10)$$

où ξ représente les cordonnées généralisées du système à savoir le déplacement horizontal du chariot z et l'angle du pendule θ et Q représente l'ensemble des forces non conservatives.

Vis-à-vis du chariot, $Q_z = F - f\,\dot{z}$: Q_z correspond à la force d'entraînement F générée par l'actionneur et transmise au chariot par la courroie à laquelle s'opposent les forces de frottements visqueux dont le coefficient est noté f.

Vis-à-vis du pendule, $Q_\theta = -k\dot{\theta}$: Q_θ correspond aux forces de frottements visqueux autour de l'axe dont le coefficient est noté k.

L'équation par rapport au chariot s'écrit :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) - \frac{\partial L}{\partial z} = F(t) - f\,\dot{z} \qquad (3.11)$$

En utilisant (3.9) et (3.11), on obtient :

$$\frac{d}{dt}(M\dot{z} + m\dot{z} - m\ell\dot{\theta}\cos\theta) - 0 = F(t) - f\,\dot{z} \qquad (3.12)$$

$$\frac{d}{dt}\left((M+m)\dot{z} - m\ell\dot{\theta}\cos\theta)\right) = F(t) - f\,\dot{z} \qquad (3.13)$$

d'où

$$\boxed{(M+m)\ddot{z}+f\dot{z}-m\ell\ddot{\theta}\cos\theta+m\ell\dot{\theta}^2\sin\theta=F(t)} \tag{3.14}$$

L'équation par rapport au pendule s'écrit :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}}\right)-\frac{\partial L}{\partial\theta}=-k\dot{\theta} \tag{3.15}$$

En utilisant (3.9) et (3.11), on obtient :

$$\frac{d}{dt}\left(-m\ell\dot{z}\cos\theta+m\ell^2\dot{\theta}+I\dot{\theta}\right)-\left(m\ell\dot{z}\dot{\theta}\sin\theta-mg\ell\sin\theta\right)=-k\dot{\theta} \tag{3.16}$$

$$-m\ell\ddot{z}\cos\theta+m\ell^2\ddot{\theta}+I\ddot{\theta}+m\ell\dot{z}\dot{\theta}\sin\theta-m\ell\dot{z}\dot{\theta}\sin\theta+mg\ell\sin\theta=-k\dot{\theta} \tag{3.17}$$

$$\boxed{\left(m\ell^2+I\right)\ddot{\theta}-m\ell\ddot{z}\cos\theta+mg\ell\sin\theta=-k\dot{\theta}} \tag{3.18}$$

Après ré-arrangement, le comportement dynamique du système est décrit par les équations suivantes :

$$\boxed{\begin{aligned}&(M+m)\ddot{z}+f\dot{z}-m\ell\ddot{\theta}\cos\theta+m\ell\dot{\theta}^2\sin\theta=F(t) &(3.19)\\ &-m\ell\ddot{z}\cos\theta+\left(m\ell^2+I\right)\ddot{\theta}+k\dot{\theta}+mg\ell\sin\theta=0 &(3.20)\end{aligned}}$$

III.2.1 - Construction du modèle non-linéaire sous Simulink

De l'équation (3.19), on peut exprimer simplement $\ddot{z}$ de la manière suivante :

$$\ddot{z}=\left(F(t)-f\dot{z}+m\ell\ddot{\theta}\cos\theta-m\ell\dot{\theta}^2\sin\theta\right)/(M+m) \tag{3.21}$$

Pour exprimer $\ddot{\theta}$, il est nécessaire d'éliminer $\ddot{z}$ de l'équation (3.20) pour éviter une boucle algébrique lors de l'exécution du schéma Simulink. On multiplie donc l'équation (3.20) par (M+m) et en remplaçant (M+m) $\ddot{z}$ par son expression (3.21), on obtient :

$$\begin{aligned}&-m\ell\left(F(t)-f\dot{z}+m\ell\ddot{\theta}\cos\theta-m\ell\dot{\theta}^2\sin\theta\right)\cos\theta+\left(m\ell^2+I\right)(M+m)\ddot{\theta}\\ &+k(M+m)\dot{\theta}+mg\ell(M+m)\sin\theta=0\end{aligned} \tag{3.22}$$

En regroupant les termes en $\ddot{\theta}$ et en divisant par $m\ell$, on obtient l'équation suivante :

$$\ddot{\theta}\left(-m\ell\cos^2\theta+\left(\ell+I/m\ell\right)(M+m)\right)=\cos\theta\ F(t)-\cos\theta\ f\ \dot{z}-m\ell\dot{\theta}^2\sin\theta\cos\theta \\ -k(M+m)\dot{\theta}/m\ell-(M+m)g\sin\theta \tag{3.23}$$

Le schéma correspondant aux équations (3.21) et 3.23) est présenté figure suivante :

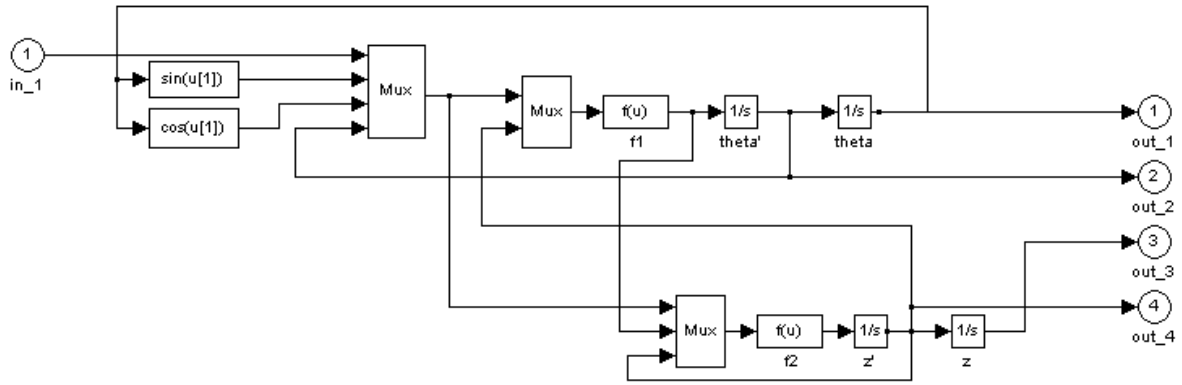


Figure III.5 : Contenu du bloc 'Modèle'

"in_1" correspond à l'entrée F(t) et out_1, out_2, out_3 et out_4 représentent respectivement les grandeurs θ, $\dot{\theta}$, z et $\dot{z}$.

"f1" et "f2" sont les deux fonctions permettant de construire $\ddot{\theta}$ et $\ddot{z}$ et dont leur contenu respectif est :

```
f1 = (u(3)*u(1)-u(3)*f*u(5)-m*l*u(4)*u(4)*u(2)*u(3)-(M+m)*k*u(4)/m/l-(M+m)*g*u(2))
/(-m*l*u(3)*u(3)+(I/m/l+l)*(M+m))
```

avec

$$u(1) \Leftrightarrow F(t),\ \ u(2) \Leftrightarrow \sin\theta,\ u(3) \Leftrightarrow \cos\theta,\ u(4) \Leftrightarrow \dot{\theta},\ u(5) \Leftrightarrow \dot{z}$$

```
f2 = (u(1)-f*u(6)+m*l*u(5)*u(3)-m*l*u(4)*u(4)*u(2))/(M+m)
```

avec

$$u(1) \Leftrightarrow F(t),\ \ u(2) \Leftrightarrow \sin\theta,\ u(3) \Leftrightarrow \cos\theta,\ u(4) \Leftrightarrow \dot{\theta},\ u(5) \Leftrightarrow \ddot{\theta},\ u(6) \Leftrightarrow \dot{z}$$

III.2.2 - Linéarisation des équations et fonction de transfert

Après avoir établi les équations non-linéaires du système, l'étape suivante consiste à effectuer une linéarisation autour des deux positions d'équilibre du système, l'une pour laquelle $\theta = 0$ correspondant à un point d'équilibre stable et l'autre pour laquelle $\theta = \pi$ correspondant à un point d'équilibre instable. Ainsi, il est possible, autour de ces points, d'effectuer un asservissement du système par des lois de commande linéaires.

En considérant des petites variations ε du pendule autour d'un point d'équilibre noté θ_o, la position θ peut s'écrire :

$$\begin{aligned}\theta &= \theta_0 + \varepsilon \\ \dot{\theta} &= \dot{\varepsilon}\end{aligned} \tag{3.24}$$

En effectuant un développement en série de Taylor à l'ordre 1 d'une fonction *f*(x) non linéaire autour de θ_o, on obtient :

$$f(\theta) \approx f(\theta_0) + \varepsilon \left.\frac{df}{d\theta}\right|_{\theta_0} \tag{3.25}$$

III.2.2.1 - Pendule en position basse (θ = 0)

Pour θ = 0, les termes non linéaires deviennent :

$$\begin{aligned}\cos\theta &\approx \cos(0) + \theta\big(-\sin(0)\big) = 1 \\ \sin\theta &\approx \sin(0) + \theta\big(\cos(0)\big) = \theta\end{aligned} \tag{3.26}$$

En introduisant ces linéarisations dans les équations du mouvement, on obtient :

$$\boxed{(M+m)\ddot{z} + f\,\dot{z} - m\ell\ddot{\theta} = F(t)} \tag{3.27}$$

$$\boxed{-m\ell\ddot{z} + \left(m\ell^2 + I\right)\ddot{\theta} + k\dot{\theta} + mg\ell\theta = 0} \tag{3.28}$$

En appliquant la Transformée de Laplace, on obtient :

$$\begin{cases}(M+m)s^2Z(s) + fsZ(s) - m\ell s^2\Theta(s) = F(s) \\ -m\ell s^2Z(s) + (m\ell^2 + I)s^2\Theta(s) + ks\Theta(s) + mg\ell\Theta(s) = 0\end{cases} \tag{3.29}$$

De ce système d'équations, on en déduit les fonctions de transfert suivantes :

$$\begin{aligned}G_1(s) &= \frac{Z(s)}{F(s)} = \frac{B(s)}{A(s)} = \frac{b_2s^2 + b_1s + b_0}{a_4s^4 + a_3s^3 + a_2s^2 + a_1s} \\ G_2(s) &= \frac{\Theta(s)}{F(s)} = \frac{B(s)}{A(s)} = \frac{c_1s}{a_4s^3 + a_3s^2 + a_2s + a_1}\end{aligned} \tag{3.30}$$

où

$$\begin{aligned}&b_2 = m\ell^2 + I \approx m\ell^2, \qquad b_1 = k, \qquad b_0 = mg\ell \\ &a_4 = (M+m)(m\ell^2 + I) - m^2\ell^2 \approx Mm\ell^2 \\ &a_3 = (M+m)k + (ml^2 + I)f \approx (M+m)k + ml^2f \\ &a_2 = (M+m)mg\ell + f\,k \\ &a_1 = mg\ell f \\ &c_1 = m\ell\end{aligned}$$

Avec les valeurs des différents paramètres présentés dans le tableau (page 9), les pôles sont :

$P_1 = 0$, $P_2 = -35$ $\rightarrow$ pôles relatifs au chariot

$P_{3,4} = -0.034 \pm 4.4j$ $\rightarrow$ pôles relatifs au pendule, complexes conjugués traduisant l'aspect oscillatoire et ici très peu amorti du pendule.

Le tracé des pôles et zéros ainsi que les diagrammes de Bode des deux fonctions de transfert sont donnés figure suivante :

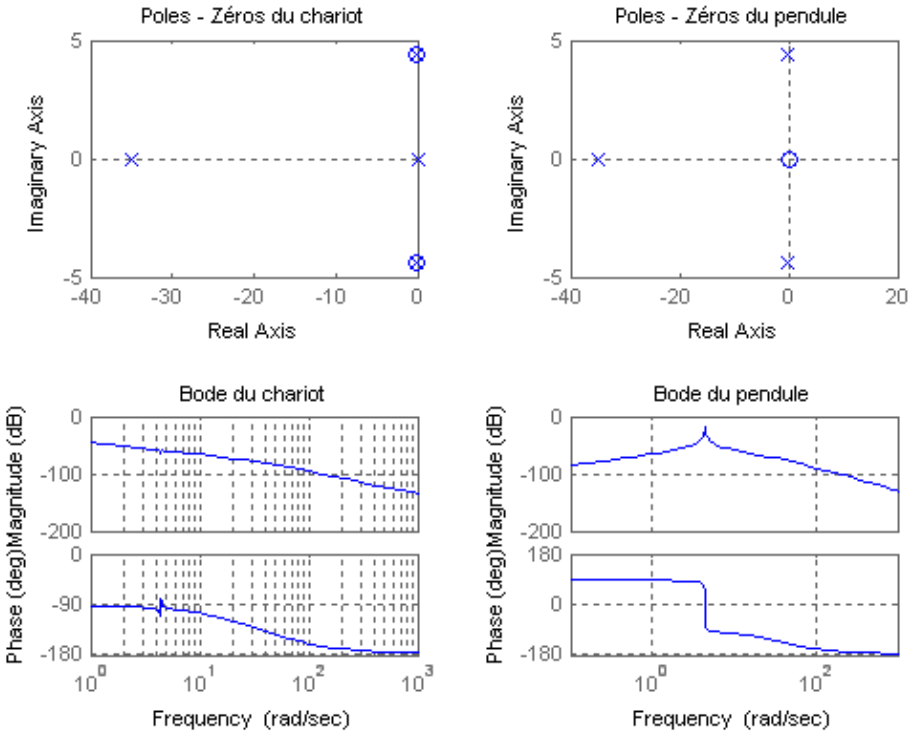


Figure III.6 : Représentation des pôles – zéros et diagramme de Bode pour le système linéarisé autour de $\theta = 0$

III.2.2.2 - Pendule en position haute ($\theta = \pi$)

Pour $\theta = \pi$, les termes non linéaires deviennent :

$$\begin{aligned} \cos\theta &\approx \cos(\pi) + (\pi - \theta)(-\sin(\pi)) = -1 \\ \sin\theta &\approx \sin(\pi) + (\pi - \theta)\cos(\pi) = \pi - \theta \end{aligned} \qquad (3.30)$$

L'analyse peut être simplifiée en définissant un nouvel angle θ' tel que : $\theta' = \theta - \pi$. Par conséquent,

$$\sin\theta \approx \pi - (\theta' + \pi) = -\theta' \qquad (3.31)$$

Cet angle θ' n'est autre que l'angle θ par rapport à la position verticale vers le haut. En considérant cette nouvelle position verticale de référence vers le haut, θ' peut se réécrire en θ.

Notons également que :

$$\dot{\theta}' = \dot{\theta} \quad \text{et} \quad \ddot{\theta}' = \ddot{\theta}$$

En introduisant ces linéarisations dans les équations du mouvement, on obtient :

$$\boxed{\begin{aligned} &(M+m)\ddot{z} + f\dot{z} + m\ell\ddot{\theta} = F(t) \\ &m\ell\ddot{z} + \left(m\ell^2 + I\right)\ddot{\theta} + k\dot{\theta} - mg\ell\theta = 0 \end{aligned}} \qquad \begin{aligned} &(3.32) \\ &(3.33) \end{aligned}$$

En appliquant la Transformée de Laplace, on obtient :

$$\begin{cases} (M+m)s^2Z(s) + fsZ(s) + m\ell s^2\Theta(s) = F(s) \\ m\ell s^2Z(s) + (m\ell^2 + I)s^2\Theta(s) + ks\Theta(s) - mg\ell\Theta(s) = 0 \end{cases} \qquad (3.34)$$

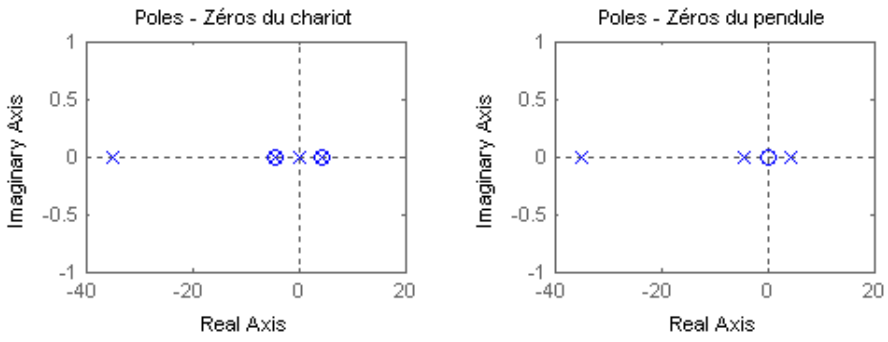
Les termes des deux fonctions de transfert (3.30) deviennent :

$$b_2 = m\ell^2 + I \approx m\ell^2, \quad b_1 = k, \quad b_0 = -mg\ell$$
$$a_4 = (M+m)(m\ell^2 + I) - m^2\ell^2 \approx Mm\ell^2$$
$$a_3 = (M+m)k + (ml^2 + I)f \approx (M+m)k + ml^2f$$
$$a_2 = -(M+m)mg\ell + f\,k$$
$$a_1 = -mg\ell f$$
$$c_1 = -m\ell$$

Avec les valeurs des différents paramètres présentés dans le tableau (page 9), les pôles sont :

$P_1 = 0$, $P_2 = -35$ $\rightarrow$ pôles relatifs au chariot
$P_3 = -4.4$, $P_4 = 4.4$ $\rightarrow$ pôles relatifs au pendule dont un est positif traduisant l'instabilité du système.

Le tracé des pôles et zéros ainsi que les diagrammes de Bode des deux fonctions de transfert sont donnés figure suivante :



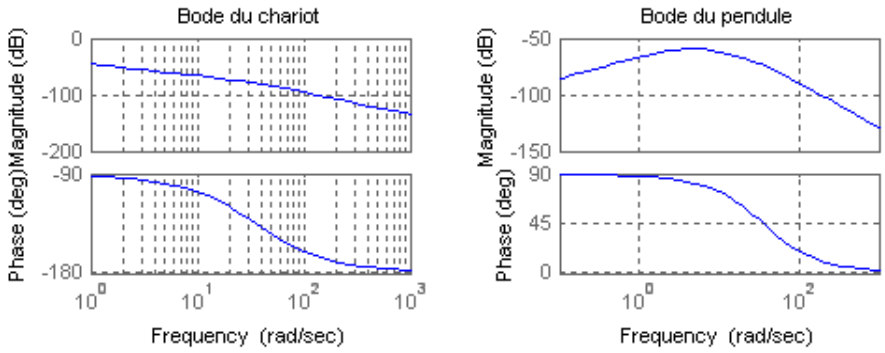


Figure III.7 : Représentation des pôles – zéros et diagramme de Bode pour le système linéarisé autour de $\theta = \pi$

III.2.3 - Représentation dans l'espace d'état

Le développement d'une commande par retour d'état nécessite la transformation des équations décrivant le mouvement du système en équations d'état. Pour cela, le vecteur d'état x(t) est défini de la manière suivante :

$$x(t) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} z(t) \\ \dot{z}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{pmatrix} \tag{3.35}$$

En se référant aux équations relatives au pendule en position verticale vers le haut (3.32) et (3.33) et en négligeant le moment d'inertie I, on obtient :

$$\begin{cases} (M+m)\dot{x}_2 + f\,x_2 + m\ell\dot{x}_4 = F(t) \\ m\ell\dot{x}_2 + m\ell^2\,\dot{x}_4 + k\,x_4 - mg\ell x_3 = 0 \end{cases} \tag{3.36}$$

A partir de la définition des variables d'état, on a :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_3 &= x_4 \end{aligned} \tag{3.37}$$

Pour obtenir le modèle d'état complet du système, il est nécessaire d'établir les expressions de $\dot{x}_2$ et $\dot{x}_4$ en fonction de l'ensemble des variables d'état. Celles-ci sont obtenues par substitution et élimination successive des termes $\dot{x}_2$ et $\dot{x}_4$ des équations (3.36). On a ainsi :

$$(M+m)\dot{x}_2 = F(t) - f\,x_2 - m\ell\dot{x}_4 \tag{3.38}$$

avec

$$m\ell^2\,\dot{x}_4 = -m\ell\dot{x}_2 - k\,x_4 + mg\ell x_3 \tag{3.39}$$

En exprimant $\dot{x}_4$ dans l'équation (3.39) et en remplaçant par son expression dans (3.38), on obtient :

$$M\dot{x}_2 = F(t) - f\,x_2 - mg\,x_3 + (k/\ell)\,x_4 \qquad (3.40)$$

D'autre part, la détermination de $\dot{x}_4$ est obtenue en remplaçant $m\ell\,\dot{x}_2$ par son expression dans l'équation (3.39).

$$\dot{x}_4 = -\frac{F}{M\ell} + \frac{f}{M\ell}x_2 + \frac{(M+m)g}{M\ell}x_3 - \frac{(M+m)k}{mM\ell^2}x_4 \qquad (3.41)$$

Le modèle linéarisé peut donc se mettre sous la forme matricielle suivante :

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) \end{aligned} \qquad (3.42)$$

avec

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -f/M & -mg/M & k/M\ell \\ 0 & 0 & 0 & 1 \\ 0 & f/M\ell & (M+m)g/M\ell & -(M+m)k/Mm\ell^2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/M\ell \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

et

$$u(t) = F(t)$$

Le même type de développement permet d'exprimer le modèle d'état du système en position verticale basse. Dans ce cas, le modèle s'écrit :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -f/M & -mg/M & -k/M\ell \\ 0 & 0 & 0 & 1 \\ 0 & -f/M\ell & -(M+m)g/M\ell & -(M+m)k/Mm\ell^2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/M\ell \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

III.2.4 - Identification de certains paramètres physiques du système

La documentation technique, fournie avec la plate forme PS600, définit numériquement un certain nombre de paramètres physiques tels que la masse du chariot, la longueur du rail, la masse et la longueur du pendule (cf. le tableau page 9) mais ne donne aucune information sur les forces de frottements par exemple.

Il a donc été nécessaire d'identifier de manière expérimentale les coefficients de frottements visqueux (f et k). Cette détermination a été réalisée en comparant la pseudo réponse impulsionnelle du système (celui-ci étant intégrateur) à celle des modèles non-linéaire et linéarisé correspondant à la position d'équilibre stable. Le signal envoyé est une "impulsion de +10 Volts pendant 0,1 seconde.

Les valeurs suivantes : f = 210 N/m/s et k = 0.003 Nm/rad/s ont permis d'obtenir les réponses impulsionnelles ci-dessous.

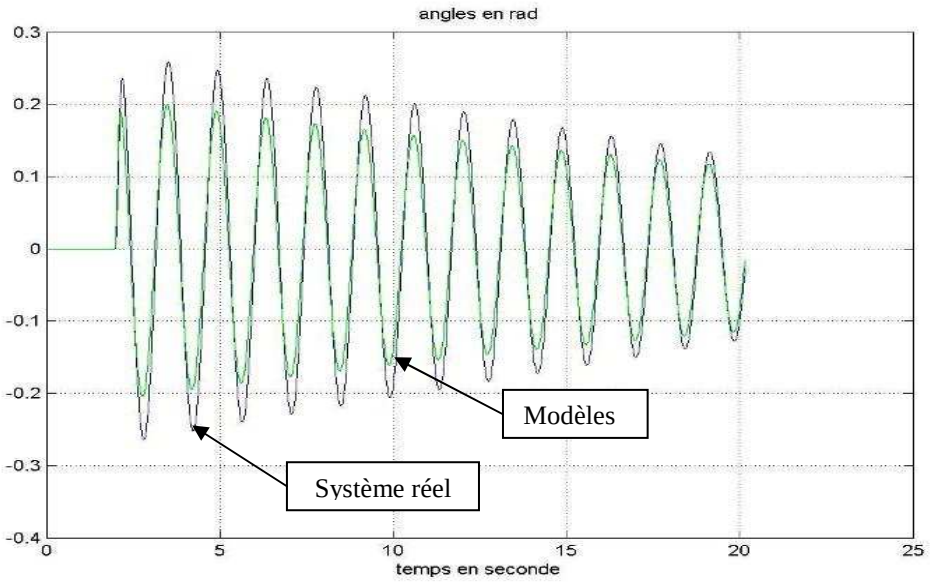


Figure III.8 : Comparaison des positions angulaires du pendule en rad autour de la position verticale basse

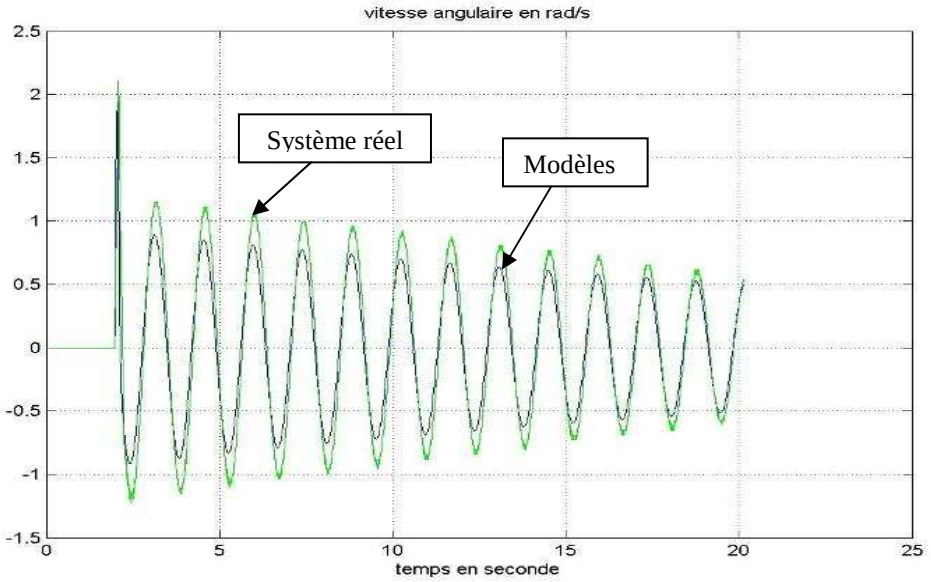


Figure III.9 : Comparaison des vitesses angulaires du pendule en rad/s autour de la position verticale basse

Avec ces valeurs, on constate que la fréquence d'oscillation est semblable entre les différentes réponses et qu'il y a une légère différence entre l'amplitude d'oscillation des deux modèles et celle du système réel.

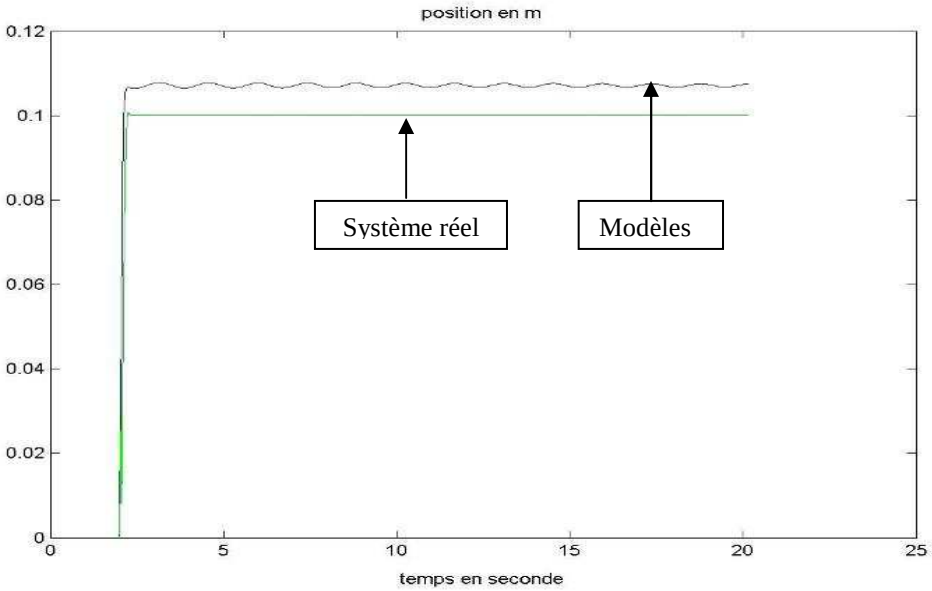


Figure III.10 : Comparaison des positions du chariot (en m) entre le système réel et les modèles

Le chariot a parcouru 10 centimètres pendant les 100 millisecondes. On observe d'une part, une erreur statique liée à l'existence de frottements secs générant une zone morte (effet non modélisé actuellement) et d'autre part, une légère oscillation traduisant l'effet du mouvement du pendule sur la position du chariot (effet non visible en réel).

Remarques :

- Les forces de frottements ne sont pas constantes sur toute la longueur du rail.

- La commande du système est effectuée en tension. L'actionneur réalise en fait une conversion tension/force. En considérant l'actionneur, non pas comme un système dynamique mais comme un simple gain, le constructeur indique la conversion suivante: ks = 22.5 N/V. La force appliquée au chariot est donc égale à : $F(t) = k_s\, u(t)$, u(t) étant une tension analogique comprise entre –10V et +10V.

- Il existe un facteur 10 entre la tension appliquée et la tension calculée sous Simulink. Une valeur de tension de +/- 1 V sous Simulink correspond à un signal de +/- 10 V qui est appliqué au moteur.

III.3 - Bloc '**Standup Routine**'

A la mise en marche, le pendule est au repos dans sa position d'équilibre stable c'est-à-dire vers le bas. La procédure de '**Standup Routine**' est destinée à amener le pendule au voisinage de sa position d'équilibre verticale instable où l'algorithme de commande va pouvoir stabiliser le pendule dans cette position verticale. Ce système doit donc permettre d'amener le pendule au voisinage de la verticale et ceci avec une vitesse angulaire suffisamment faible. (cf. annexe IV)

III.4 - Bloc 'Commande'

La structure du schéma général développé à partir de l'environnement Simulink a été conçue de telle manière à être complètement adaptable à l'intégration d'un nouveau correcteur en modifiant uniquement le bloc 'Commande' du schéma général. A titre d'illustration, on présente une commande par retour d'état : la commande LQ.

La mise du système sous forme d'état (équation 3.42) conduit naturellement à mettre en place une commande par retour d'état. Celle-ci permet de stabiliser les modes commandables d'un système et de les placer arbitrairement dans une région stable du plan complexe (1/2 plan droit). Différentes stratégies peuvent être développées. Compte tenu de l'incertitude de certains paramètres du modèle, le choix a été de synthétiser une commande robuste garantissant la stabilité robuste du système en boucle fermée. La stratégie de commande utilisée est la méthode de la commande Linéaire Quadratique (LQ) basée sur la minimisation du critère quadratique J suivant :

$$J = \int_0^{\infty} \left(x^T Q\, x + u^T R\, u \right) dt \qquad (3.43)$$

A travers ce critère, on désire ramener l'état du système à l'origine en appliquant une commande dite optimale d'énergie bornée. Dans ce critère, les matrices constantes Q et R sont symétriques et positives ($Q > 0$ et $R > 0$).

La matrice R pondère la commande et permet de régler le compromis que l'on désire effectuer entre un retour rapide de l'état du système à zéro (précision désirée) et l'énergie dépensée pour le faire. La matrice Q permet de choisir les états que l'on désire ramener rapidement à zéro.

La commande qui minimise le critère quadratique J est une loi de retour d'état qui s'exprime sous la forme:

$$u(t) = -K\, x(t) \quad \text{avec} \quad K = R^{-1} B^T P \qquad (3.44)$$

où P est l'unique solution symétrique définie positive de l'équation de Riccati suivante :

$$PA + A^T P - P B R^{-1} B^T P + Q = 0 \qquad (3.45)$$

On a alors :

$$J = x_0^T P\, x_0 \quad \text{où } x_0 \text{ est l'état initial.}$$

Remarques :

- On peut généraliser la commande LQ dans le cas où la référence n'est pas nulle. Une méthode consiste alors à élaborer une commande avec un pré-compensateur. Dans le cas du pendule, la position angulaire étant définie à partir de la position verticale basse ($\theta = 0$), l'asservissement du pendule en position verticale haute a été réalisé en introduisant un offset sur l'angle d'un facteur π (cf. figure III.11). On s'est donc ramené à un problème de régulation autour de 0 ce qui évite l'introduction d'un pré-compensateur.

- Lorsque la paire (A, B) est commandable et la paire (A, C) est observable, la solution définie positive de l'équation de Riccati existe toujours.

- Dans les mêmes conditions, la matrice d'état en boucle fermée obtenue (A – BK) a toutes ses valeurs propres à partie réelle négative.

L'implémentation du correcteur sous Simulink est présentée figure suivante :

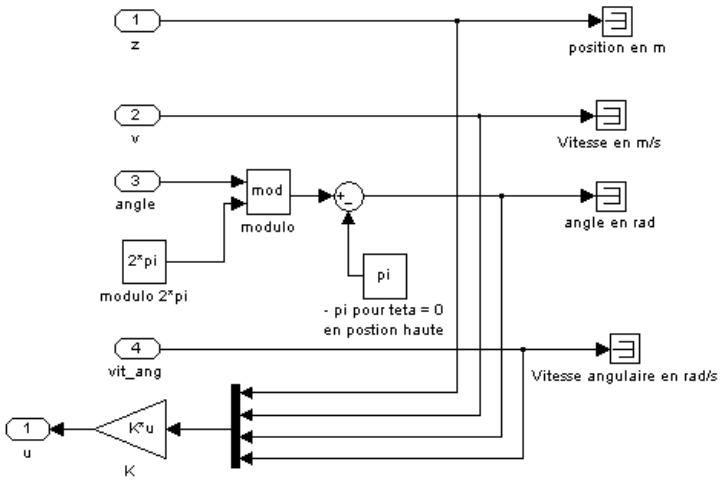


Figure III.11 : Contenu du bloc 'Commande'

Les résultats expérimentaux sont présentés ci-après, ils ont été obtenus avec un correcteur LQ calculé avec les pondérations suivantes :

$$Q = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0{,}1 \end{bmatrix} \quad \text{et} \quad R = 10$$

Les difficultés majeures rencontrées dans la définition de ces pondérations ont été les suivantes :

- Le déplacement du chariot sur le rail est limité.
- L'amplitude du débattement angulaire ne doit pas trop importante.
- Le signal de commande u (t) doit pas être admissible par l'actionneur.

Le contenu de la matrice de pondération Q montre que l'on ne s'est pas préoccupé de limiter les vitesses atteintes par le chariot.

Le correcteur K a été calculé avec la fonction '**lqr**' de Matlab. Les valeurs propres obtenues pour le système en boucle fermée sont :

[-35.8311, -4.4615, -4.2923 -0.6782]

Ces valeurs sont à comparer avec les pôles du système en boucle ouverte (page 15).

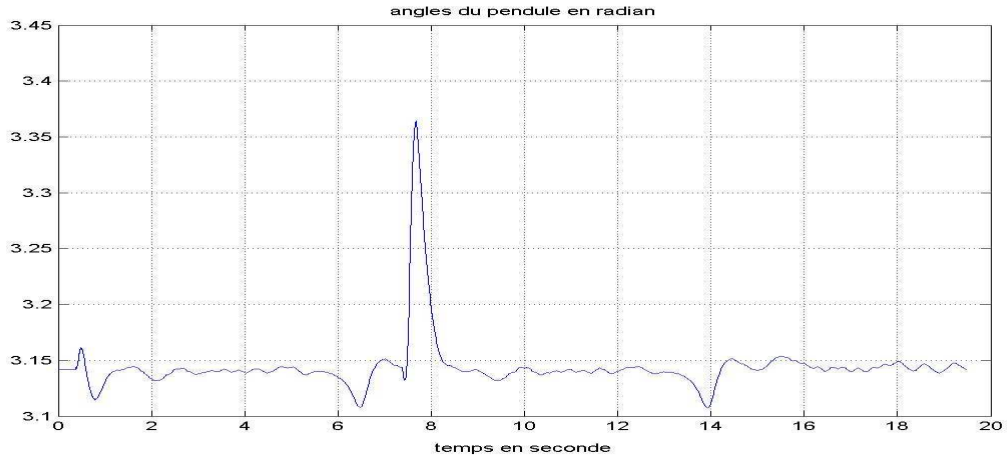


Figure III.12 : Position du pendule (en rad)

A l'instant 7,8 s, le pendule a été écarté de sa position verticale de manière à apprécier les performances de la boucle fermée en régulation.

La figure III.13 montre le déplacement du chariot. Il s'écarte au total d'environ 40 cm autour de sa position initiale (de -15 cm à 25 cm).

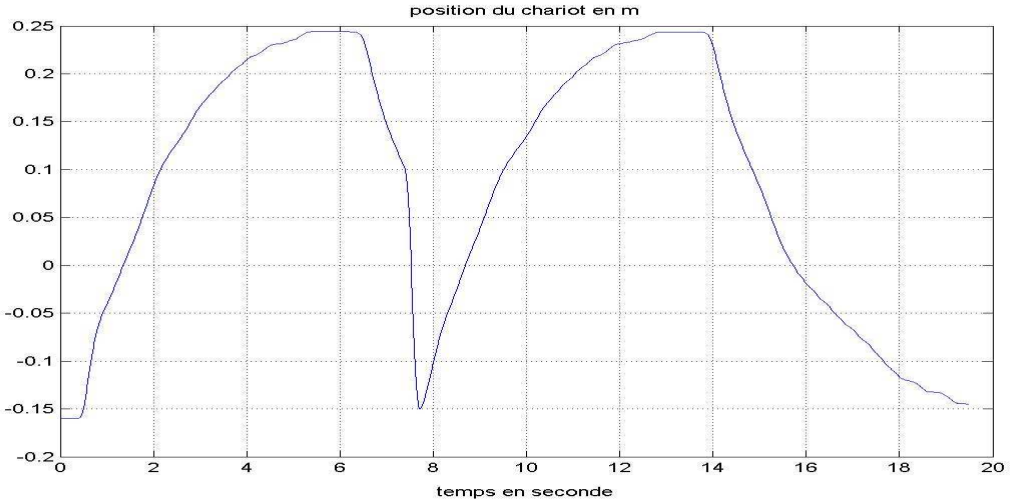


Figure III.13 : Position du chariot (en m)

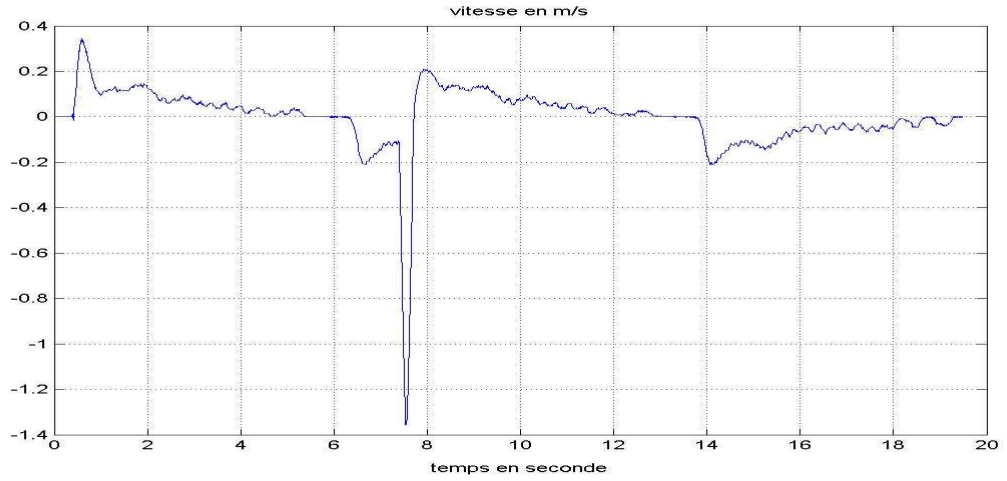


Figure III.14 : Vitesse du chariot (en m/s)

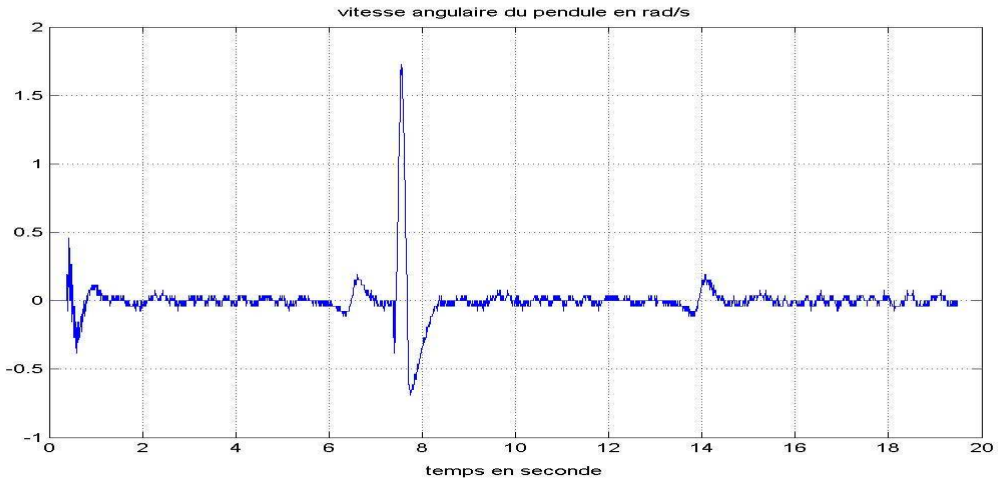


Figure III.15 : Vitesse angulaire (en rad/s)

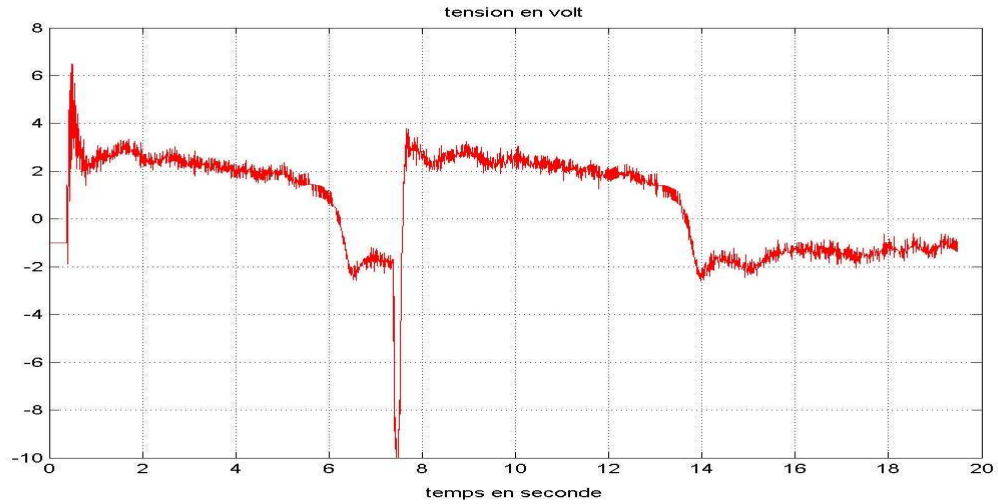


Figure III.16 : Commande (en Volts)

Hormis une légère saturation (- 10 volts) au moment de l'application de la perturbation, la commande reste comprise entre -10 et +10 volts.

ANNEXES

ANNEXE I : Modes de pilotage

Pilotage manuel (repère ①)

Pour piloter la maquette en mode manuel, il faut maintenir l'interrupteur monostable situé en face avant du bloc de commande en position haute et faire varier la tension de sortie en tournant le potentiomètre qui est à coté de l'interrupteur. On peut ainsi bouger le chariot de droite à gauche et réciproquement. On peut utiliser ce mode de pilotage principalement pour positionner le chariot jusqu'à la position de référence (le zéro).

Figure A.I.1 : Façade du bloc électronique de commande

Pilotage externe (repère ②)

Piloter la maquette en mode externe signifie que l'on va fournir une tension de commande à l'aide d'un générateur externe pour déplacer le chariot. Il faut pour cela enclencher l'interrupteur bistable en face avant du bloc de commande et fournir une alimentation |U|<10V sur les entrées BNC situées à coté de l'interrupteur.

Pilotage automatique

Le véritable intérêt de cette maquette est de pouvoir être piloté à distance en mode automatique. Pour fonctionner en mode automatique, il faut au préalable débloquer le système de puissance.

Ce déblocage est fait à distance en envoyant deux signaux particuliers à des fréquences précises sur le bloc de commande (DOUT1 et DOUT2 dans le bloc Digital I/O du schéma Simulink). La figure suivante décrit les signaux à envoyer.

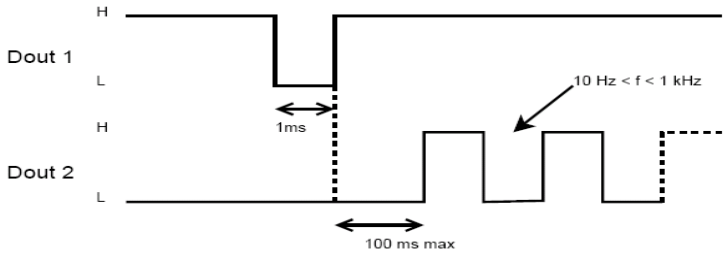


Figure A.I.2 : Signaux de déblocage du système de puissance

La réalisation sous Simulink de ces signaux est donnée ci-après.

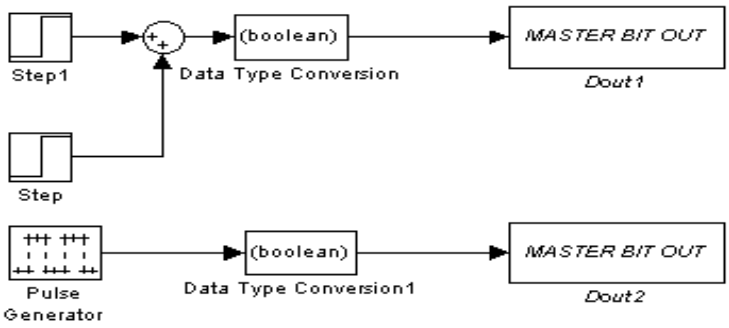


Figure A.I.3 : Réalisation sous Simulink des signaux de déblocage (contenu dans le bloc Digital I/O)

On rappelle que la période d'échantillonnage a été fixée à 0,01 s.

ANNEXE II : Carte dSPACE 1104

La carte DSP utilisée est la DSPACE DS1104. Le processeur principal est un MPC8240, avec un cœur PowerPC 603^{e} et une horloge interne à 250 MHz. Il a une capacité mémoire de 8 Mo en Flash et de 32 Mo en SDRAM.

Il dispose de 8 convertisseurs analogiques numériques (4 en 16 bits, 4 en 12 bits), de 8 convertisseurs numériques analogiques (CNA) de 16 bits pouvant délivrer une tension analogique comprise entre -10V et +10V, d'une liaison série, de 2 codeurs incrémentaux, de 20 entrées-sorties numériques, d'un DSP esclave (TMS320F240) et de 3 timers (32 bits) pouvant fonctionner de manière indépendante.

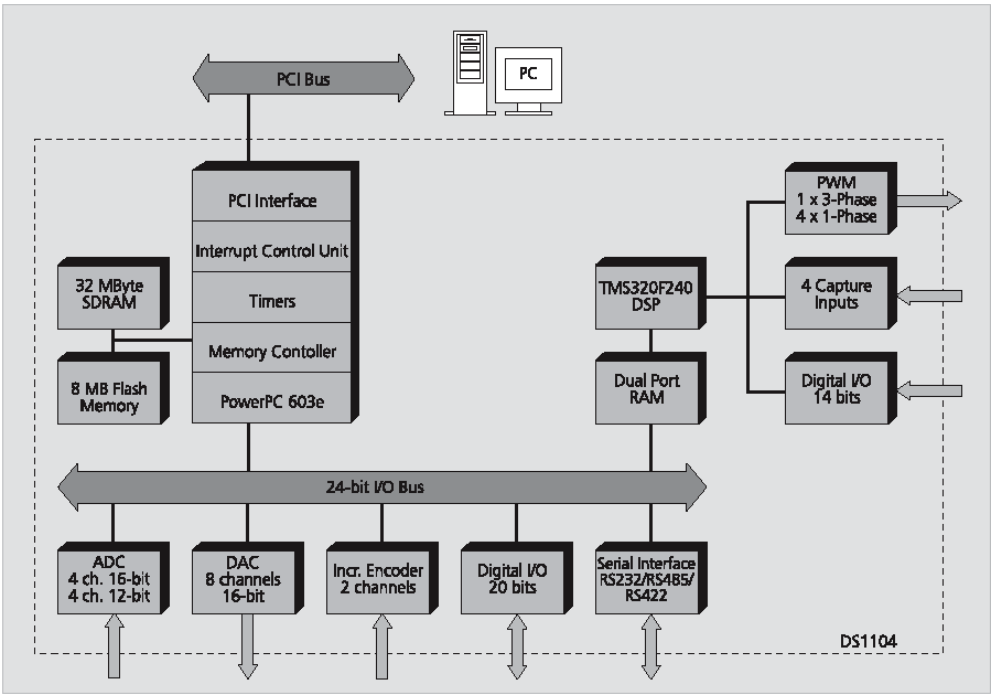


Figure A.II.1 : Architecture du DS1104

Pour programmer le DSP, il faut réaliser tout d'abord un schéma dans l'environnement Simulink de Matlab. Il est préférable de se placer dans son répertoire de travail pour élaborer le fichier Simulink car la compilation génère de nombreux fichiers contenus dans un nouveau répertoire généré à chaque étape de compilation.

Dans "Simulink Library Browser" on trouve une librairie nommée "dSPACE RTI1104" dans laquelle on peut choisir les composants utilisés pour la réalisation du schéma Simulink (Master Bit in/out, set encoder,…).

Un exemple de blocs utilisés est présenté figure A.II.2. En faisant un double click sur chaque bloc, on peut choisir le champ dans lequel on souhaite écrire ou lire les données.

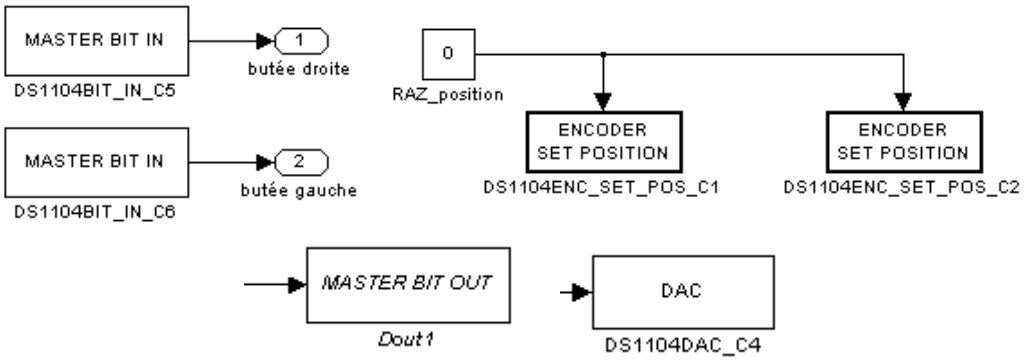


Figure A.II.2 : Exemple de blocs utilisés pour la conception du schéma Simulink

Une fois le schéma Simulink terminé et sauvegardé, l'étape suivante consiste à générer le code associé au schéma Simulink et à le transférer dans le DSP. Si la compilation a réussi, le chargement du programme et son exécution dans le DSP se font automatiquement. Pour compiler un schéma Simulink :

- Aller dans le menu **simulation**, puis **simulation parameters.**
- Dans l'onglet **Solver,** fixer **Type** à **fixed-step** puis **fixed step** à Ts (période d'échantillonnage choisie). Mettre **stop time** à inf.
- Dans l'onglet **real-time workshop**, choisir la cible puis **Build.**

La première compilation d'un schéma est assez longue (environ 30 à 40 secondes selon la puissance des PC car d'autres librairies sont compilées et la connexion avec la carte doit être établie). Ensuite, les compilations suivantes sont plus rapides (environ 10 secondes).

La compilation génère de nombreux fichiers dont deux sont particulièrement importants :

- **Schéma.sdf** listant tous les paramètres du schéma bloc, utilisés sous Control Desk pour faire le lien entre le firmware et l'IHM.
- **Schéma.ppc** qui est le firmware chargé dans la carte DSP.

NB : Lorsque le programme est compilé, il utilise par défaut les valeurs des blocs du schéma Simulink car celui-ci est automatiquement exécuté. Attention à bien choisir les valeurs initiales de ces blocs.

Le logiciel Control Desk permet la réalisation d'une interface graphique permettant de visualiser et de modifier en temps réel les différentes variables du schéma Simulink. Les extensions utilisées sous Control Desk sont :

Schema.cdx : contient le projet ou experiment
Schema.csv : fichier d'acquisition de mesures au format ASCII
Schema.idf : fichier résultant de l'acquisition des mesures en mode Stream to Disk
Schema.lay : contient le panneau graphique développé (c'est la layout)
Schema.mat : fichier d'acquisition de mesures pouvant être chargées sur Matlab
Schema.ppc : contient l'application (donc l'exécutable) pouvant être chargée dans le DSP.
Schema.sdf : contient l'ensemble des paramètres du fichier Simulink

ANNEXE III : Control Desk

Control Desk est une interface qui permet de visualiser en temps réel différentes variables du fichier développé sous Simulink et de modifier également des paramètres définissant le mode de fonctionnement des blocs constituant le schéma Simulink. La visualisation de variables ou de signaux et la modification de paramètres sont possibles par l'intermédiaire d'instruments graphiques que l'on sélectionne. Un exemple de construction d'un écran graphique est présenté ci-après (ou IHM : Interface Homme Machine).

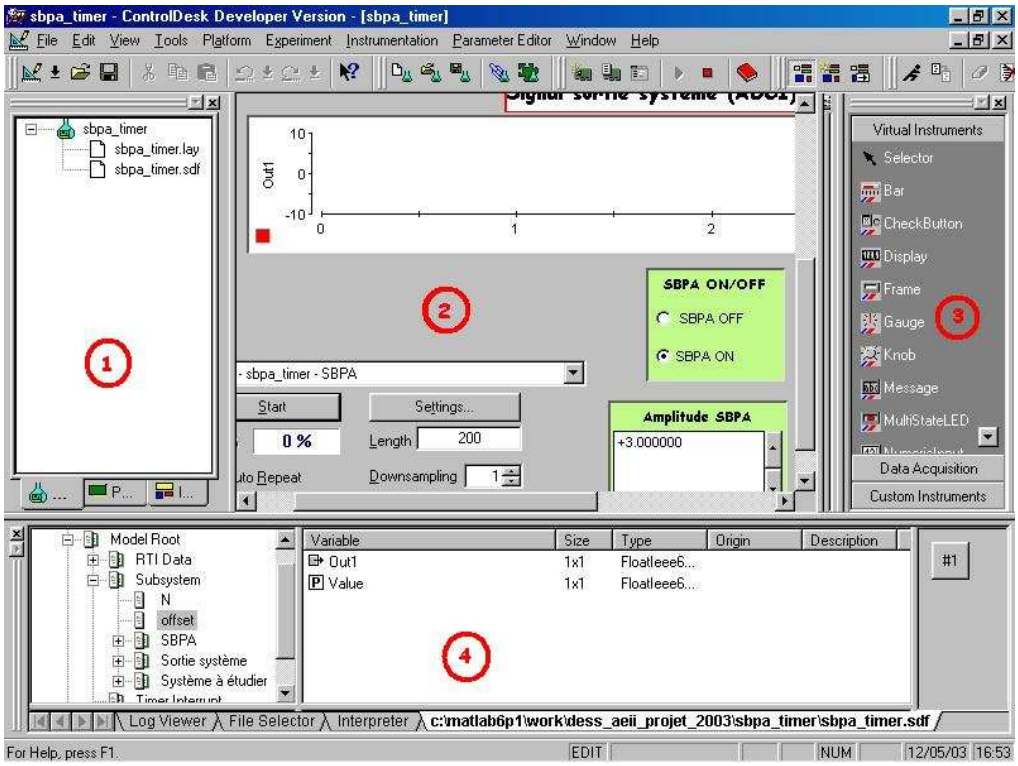


Figure A.III.1 : Exemple d'IHM développée sous Control Desk

Les différentes fenêtres rencontrées sur Control Desk sont :

1 – la fenêtre de navigation (Navigator).
2 – l'aire de travail contenant la layout.
3 – le choix d'instrument (instrument selector).
4 – la fenêtre d'outils (tool window).

Si l'une de ces fenêtres n'est plus visible à l'écran, on peut les ouvrir en sélectionnant l'onglet: View → Controlbar.

Démarrer un nouveau projet sous Control Desk

Sous Control Desk, un projet s'appelle un "experiment".

Dans l'onglet 'File', cliquer sur 'New experiment' puis saisir le nom et le répertoire désiré pour le projet. Dans l'exemple, le fichier s'appelle essai.cdx.

Remarque : un "layout" est une interface graphique à laquelle on peut ajouter divers instruments dans le but de visualiser ou de modifier en temps réel les différentes variables du projet.

Dans 'File', cliquer sur 'New' → 'Layout'
↳Se référer à la partie créer un Layout

Pour lancer la simulation, cliquer sur l'icône . On est alors en mode simulation (ou animation).

Ouvrir un projet existant

Pour ouvrir un projet existant, aller dans 'File', cliquer sur 'Open experiment'. Charger l'application dans la carte DSP.

Par exemple, l'application "essai" peut être chargée dans le DSP en ouvrant l'expérience essai.cdx puis en déplaçant essai.ppc jusqu'à la fenêtre Navigator comme le montre la figure A.III.2. Dans ce cas, il n'est pas nécessaire de compiler le schéma Simulink.

On peut alors lancer la simulation en cliquant sur l'icône et observer les différentes variables ou signaux en temps réel.

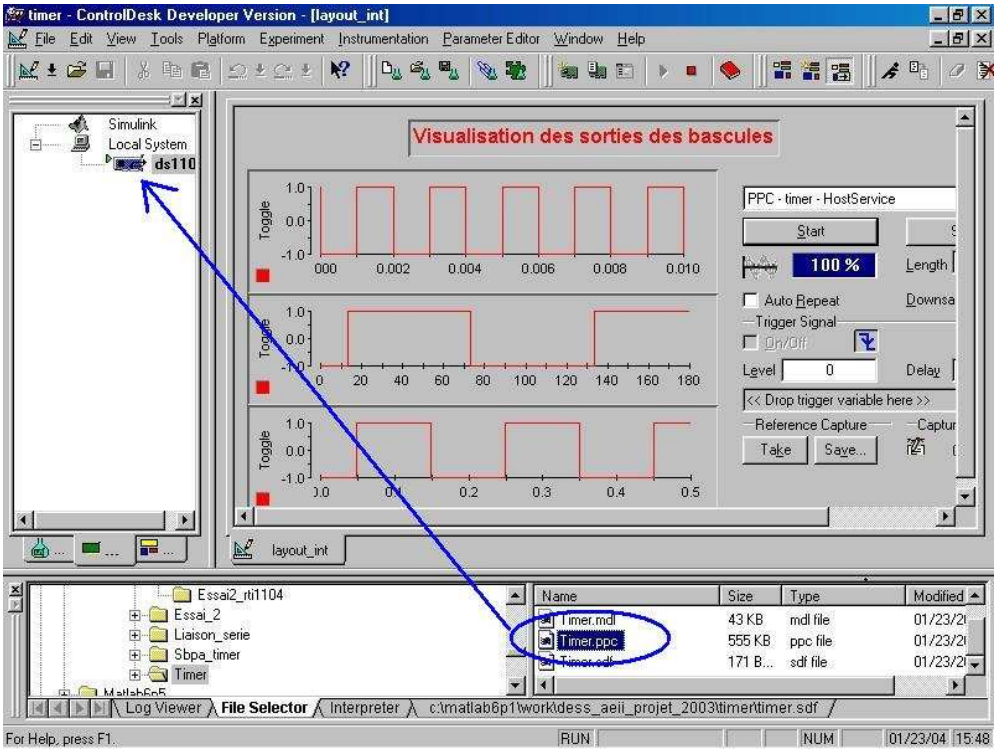


Figure A.III.2 : Exemple de chargement d'une application dans la carte DSP

Travailler dans l'espace de travail

- **Créer un layout**

L'espace de travail est l'endroit dans lequel on va créer un layout. Il existe deux catégories d'instruments :

- *Virtual instruments* : ces instruments permettent la visualisation et/ou la modification de variables.
- *Data acquisition* : ces instruments permettent l'acquisition de variables (à l'écran ou dans un fichier).

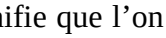
Le logiciel offre également la possibilité grâce à l'onglet 'Custom instruments' d'éditer de nouveaux instruments et de les sauvegarder dans une librairie.

Au départ, le layout est une fenêtre grise. Pour développer son propre layout, il faut se mettre en mode Edit (icône).

Remarque : on peut réaliser plusieurs layouts dans un même projet. On passe de l'un à l'autre en cliquant sur l'onglet 'Layout' de la fenêtre de gauche et en choisissant le layout à ouvrir.

Il faut ensuite ajouter tous les instruments de visualisation et de modification de variables. Enfin, il faut affecter à chaque bloc sa variable et modifier les propriétés de l'instrument si besoin, en double cliquant dessus.

Dans la fenêtre d'outils, on récupère les différentes variables ou paramètres du schéma Simulink. L'initialisation d'un objet graphique est réalisée en effectuant un clic gauche sur la variable et en la faisant glisser jusqu'à l'instrument désiré. Le sigle

signifie que l'on va pouvoir visualiser la variable. Le sigle 

signifie que l'on va accéder à la valeur de la variable et pouvoir la modifier.

L'instrument 'Capture Settings' dans 'data acquisition' est un outil essentiel, il permet de choisir la durée de l'acquisition, de l'arrêter momentanément, de sauvegarder une acquisition de données.

Le bloc 'PlotterArray' dans 'data acquisition' permet la visualisation sous forme graphique de la variable au cours du temps.

Pour de plus amples informations, une description détaillée des instruments est donnée pages 175-178 du livre "Experiment Guide" de la documentation dSPACE.

Pour modifier les propriétés d'un instrument, double-cliquer dessus puis sélectionner 'Properties'. On peut ainsi modifier le nom d'une variable, régler l'échelle des axes, etc.

- **Sauvegarder des données sur fichier**

Control Desk offre la possibilité d'effectuer une acquisition de données. L'outil 'Capture Settings' dans 'data acquisition' dispose d'un onglet 'capture variable'. On peut effectuer des acquisitions simples, avec sauvegarde automatique, acquisitions continues, etc.

On peut effectuer des sauvegardes au format .csv ou .mat. Le premier permet une sauvegarde dans un fichier ASCII éditable directement depuis ControlDesk. Le deuxième présente l'avantage d'être directement récupérable sur Matlab sous forme d'une structure. Après avoir ouvert le fichier .mat, on peut tracer les différentes variables. On accède à la variable temps par la commande **<nomdufichier>.X.Data** et aux variables observées par **<nomdufichier>.Y(n).Data**.

Dans l'outil 'Capture Settings', cliquer sur 'Settings' puis sur l'onglet 'Acquisition' afin de sélectionner le mode de capture désiré :

Simple : fait l'acquisition du signal pendant la durée établie (length) dans 'Settings' (de 0 à 100% du temps capturé).

Autosave : fait une capture de données dans le fichier spécifié.

Autoname : effectue une capture dans le fichier spécifié et incrémente le nom du fichier automatiquement. Exemple : defaut001.mat, defaut002.mat …

Continuous : permet de capturer les données de manière continue jusqu'à ce que l'on stoppe l'acquisition. La dernière période visualisée est alors sauvegardée. Vérifier que la vitesse du processeur temps réel soit compatible avec la vitesse d'acquisition du PC.

Stream to disk : enregistre les données en continu et les écrits sur le disque dur. Cette acquisition crée des fichiers portant l'extension '.idf'. A la fin de l'acquisition, il faut convertir le fichier '.idf' en un fichier '.mat'. Pour cela, aller dans Tools, cliquer sur Convert IDF File. Lorsque l'acquisition est arrêtée, on peut visualiser les données depuis le début.

La conversion peut se faire de deux manières différentes :

- Conversion vers un fichier .mat contenant toutes les données.
- Conversion vers n+1 fichiers .mat contenant chacun les données temporelles d'une variable plus un fichier temporel.

Sous Matlab, on charge le fichier .mat (**load <nom_du_fichier>.mat)**. Ensuite, on trace les différentes variables en récupérant le temps par la commande **<nomdufichier>.X.Data** et la (les) variable(s) observée(s) par **<nomdufichier>.Y(n).Data**. Pour faciliter ce travail un fichier **prog.m** a été réalisé (cf. ci-après).

Programme Matlab de récupération des données de Control Desk (prog.m)

```
load essai.mat;
t = essai.X.Data;
var1 = essai.Y(3).Data;
var2 = essai.Y(2).Data;
var3 = essai.Y(1).Data;

%tracé graphique

figure;
plot(t, var1, t, var2, t, var3);
grid;
title('Mettre un titre');
XLABEL('Temps en secondes');
YLABEL('Variables visualisée');
```

La fenêtre d'outils

La fenêtre d'outils comprend 'Log viewer', 'Interpreter', 'File selector' et le fichier de variable portant l'extension '.sdf'.

Log viewer: affiche les messages générés par Control Desk ou par la carte temps réel.

File selector: permet de sélectionner une application et de pouvoir la télécharger dans la carte DSP. Ceci évite d'avoir à effectuer la commande Build à partir du schéma Simulink lorsqu'une version compilée existe déjà.

Fichier de variable: (extension .sdf) permet d'accéder aux variables de l'application.

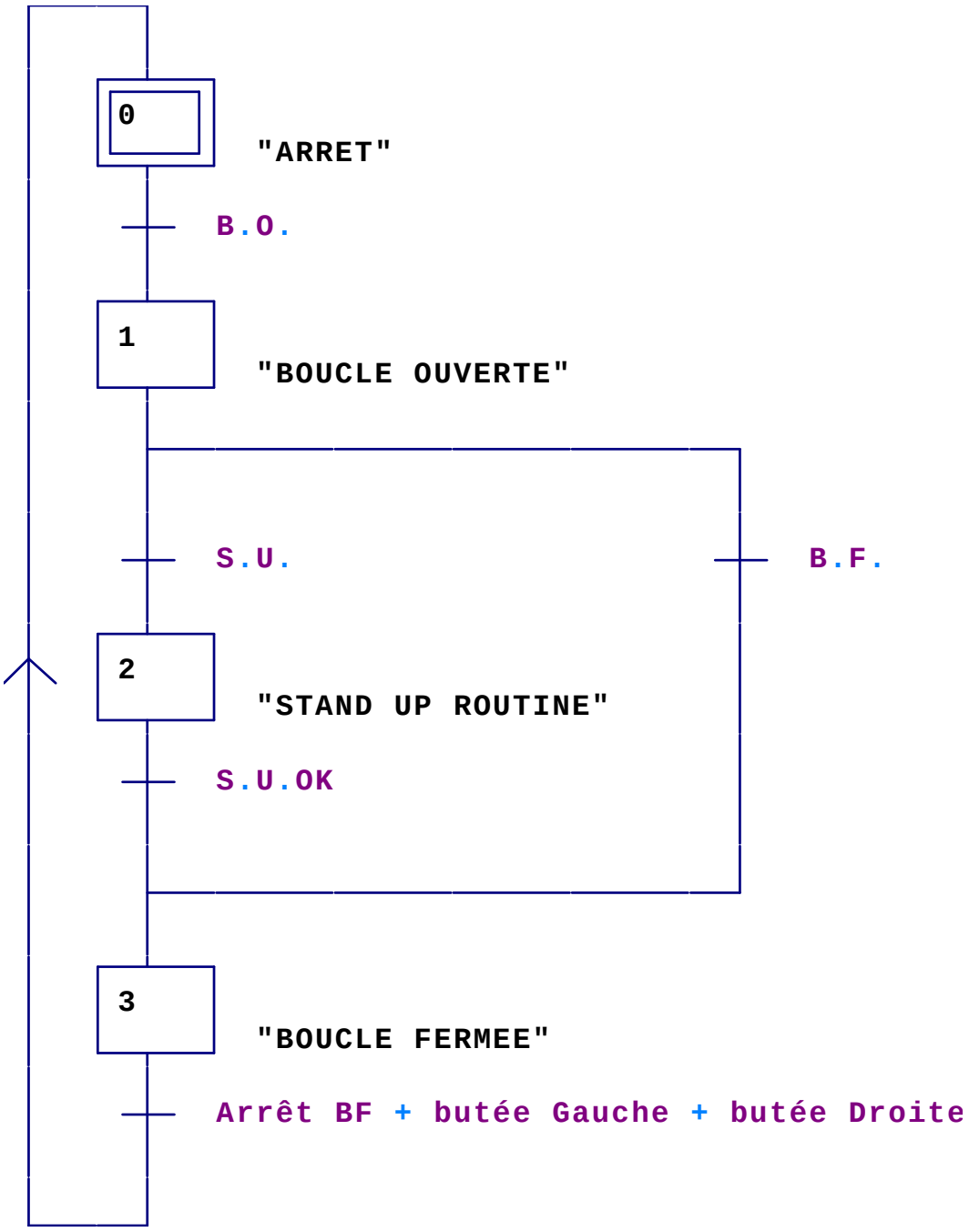
Le fichier de variables est chargé automatiquement dans le projet lors de l'exécution de la commande Build de Simulink.

Si toutefois le fichier de variables n'est pas chargé, aller dans File → Open variable file et charger le fichier portant l'extension .sdf qui est dans le répertoire de travail.

A partir du fichier de variables, on peut sélectionner une variable ou la valeur d'une variable afin de la faire glisser dans un instrument du Layout.

ANNEXE IV : Modes de marche

Représentation grafcet des Modes de marche du pendule inversé



ANNEXE V : Standup routine

La stratégie de base est de déplacer le chariot de telle sorte que l'énergie soit ajoutée graduellement au pendule. Cela nécessite d'effectuer un asservissement de position du chariot. La position de référence à atteindre va dépendre de l'angle du pendule. D'autre part, étant donné que la course du chariot est de longueur limitée, la trajectoire du chariot doit subir des phases d'accélération et de ralentissement. Afin d'utiliser au maximum la quantité de travail positive fournie au pendule, la trajectoire doit être telle que le chariot accélère lorsqu'il y a un travail important (angle du pendule au voisinage de ± 90°) et ralentisse lorsqu'il y a peu de travail (angle du pendule au voisinage de 0°). On peut ainsi définir des secteurs correspondants à des phases d'accélération ou de ralentissement.

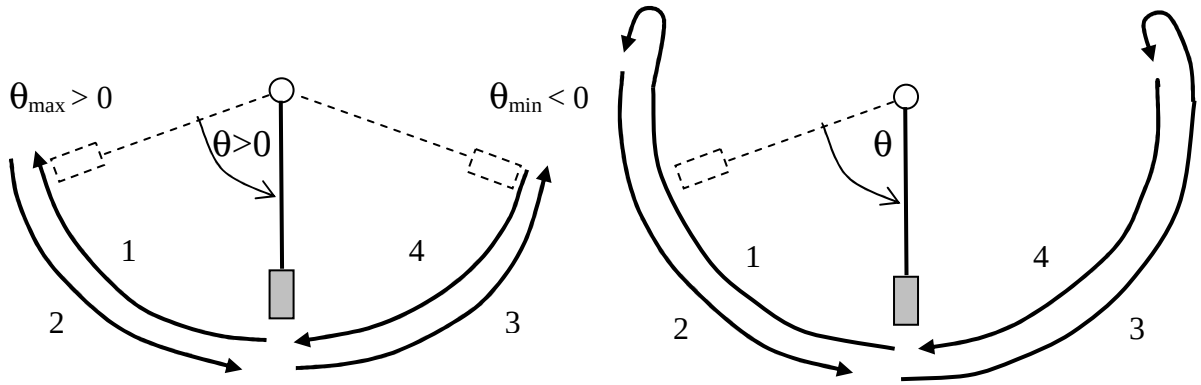


Figure A.V.1 : secteurs de balancement du pendule, faible amplitude (à gauche), forte amplitude (à droite)

Sur la figure précédente, les secteurs 2 et 4 correspondent à des régions actives dans lesquelles le chariot doit être déplacé afin d'accroître l'énergie du système. Par contre, les secteurs 1 et 3 sont des régions dans lesquelles le chariot ne doit pas bouger.

Il est nécessaire de construire une table donnant la consigne de position du chariot (en mètres) en fonction de l'angle du pendule (en degrés). Plus le pendule va haut (plus l'angle augmente), plus l'énergie fournie au pendule doit diminuer (plus la consigne de position doit être petite) afin d'approcher de la position verticale avec une vitesse angulaire décroissante.
Les variables θ_{max} et θ_{min} sont utilisées alternativement, pour ce calcul, selon le sens de balancement du pendule. θ_{max} correspond à la valeur maximale de l'angle du pendule obtenue lors du précédent balancement à gauche et θ_{min} correspond à la valeur maximale de l'angle du pendule obtenue lors du précédent balancement à droite.

Le processus de relèvement se décompose en trois phases :

- $1^{ère}$ phase : envoi d'une consigne de position du chariot, en échelon, d'un côté puis de l'autre, afin de démarrer le balancement du pendule (cf. routine 1).

- 2ème phase : génération de signaux de consignes doux afin d'augmenter de manière efficace le balancement du pendule jusqu'à l'obtention d'un angle de ± 100°(cf. routine 2).

- 3ème phase : génération de signaux de consignes d'amplitude limitée afin d'approcher de la position verticale avec une vitesse angulaire faible(cf. routine 3).

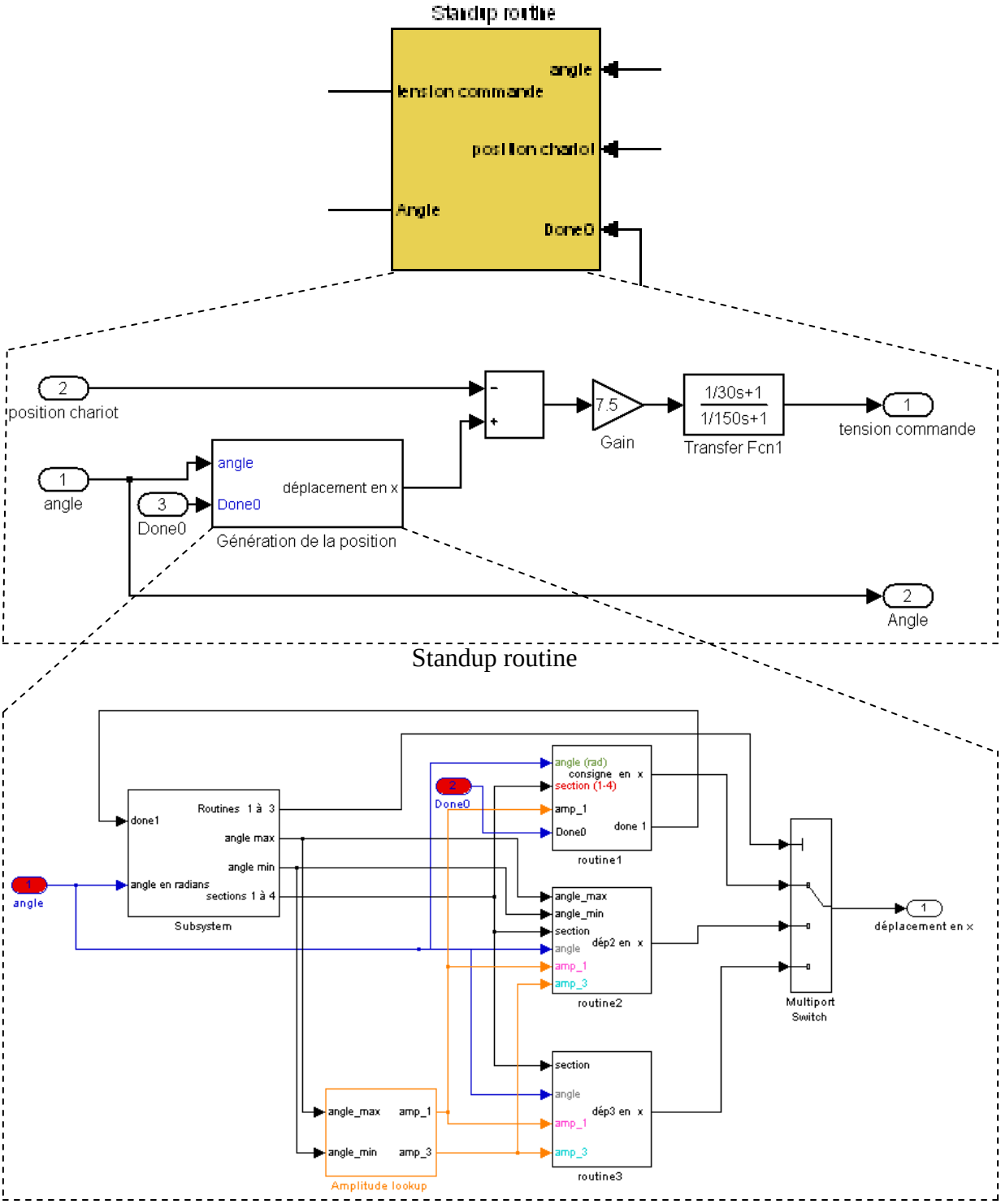


Figure A.V.2 : Contenu du bloc 'Standup Routine'

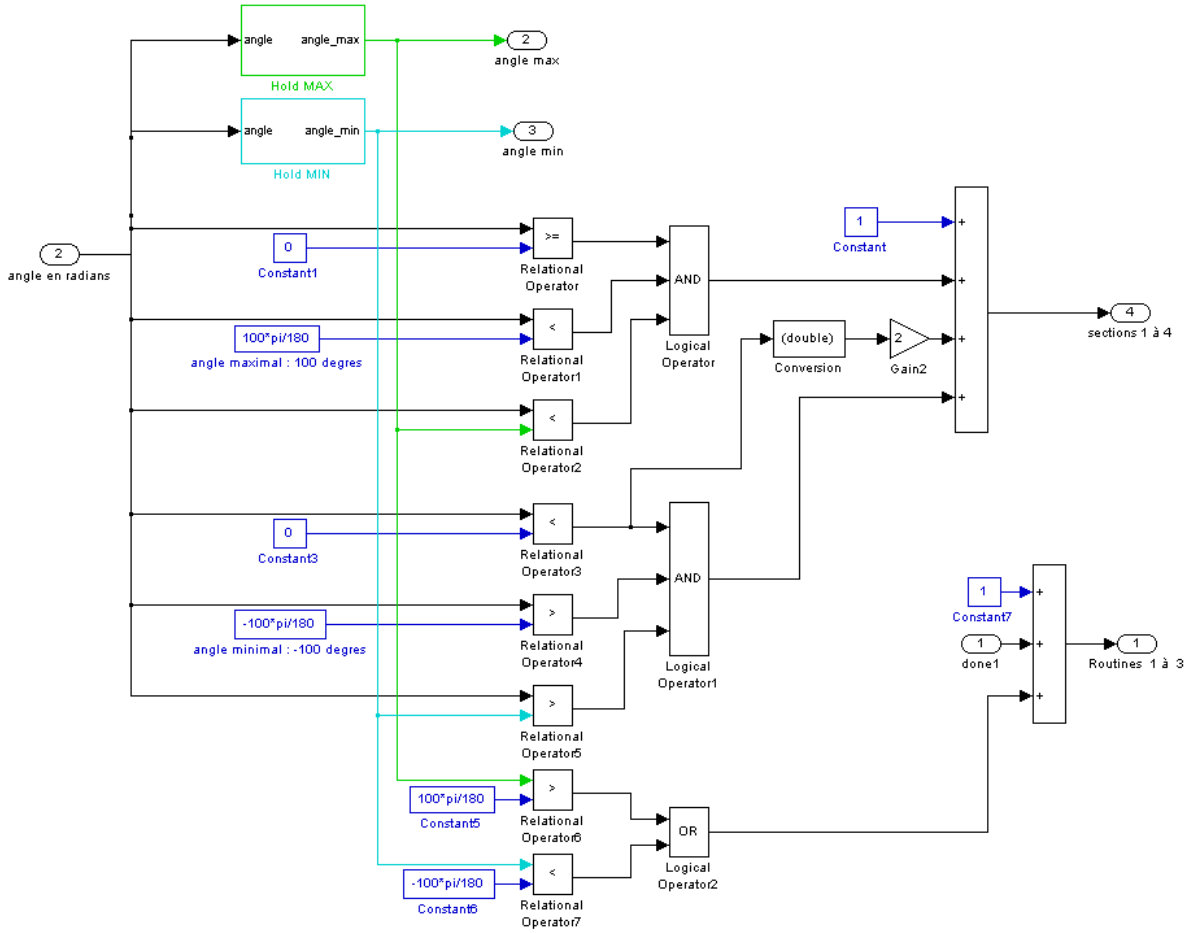


Figure A.V.3 : Contenu du bloc 'Subsystem'

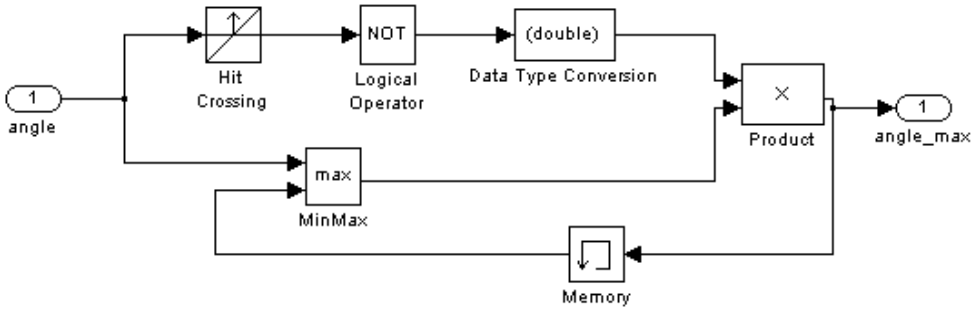


Figure A.V.4 : Contenu du bloc 'Hold MAX', détection de l'angle maximal (à gauche) lors du dernier balancement

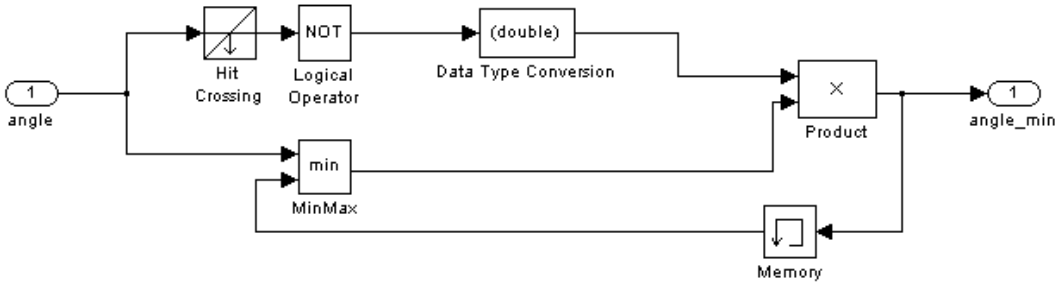


Figure A.V.5 : Contenu du bloc 'Hold MIN', détection de l'angle maximal en valeur absolue (à droite) lors du dernier balancement

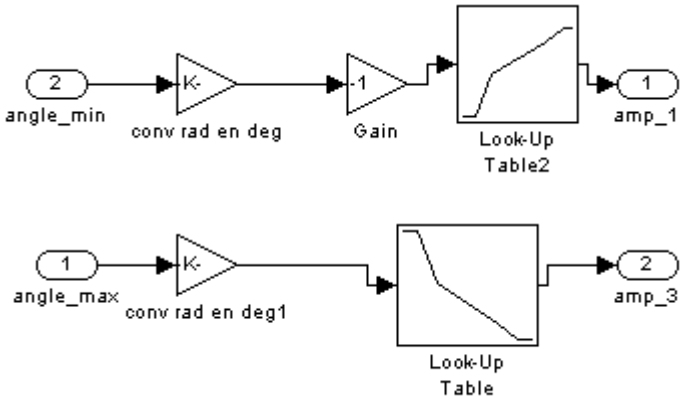


Figure A.V.6 : Contenu du bloc 'Amplitude lookup', détermination de la consigne de position du chariot en fonction de la valeur maximale de l'angle du pendule (en valeur absolue) atteinte lors du dernier balancement

Avec K = 180/π (conversion de degrés en radians).

- ✓ Look-Up Table2

θ_{min} (°) : [0 ; 20 ; 50 ; 145 ; 175 ; 190]
Amp_1 (m): [-0.25 ; - 0.25 ; -0.15 ; -0.07 ; -0.04 ; -0.04]

- ✓ Look-Up Table

θ_{max} (°) : [0 ; 20 ; 50 ; 140 ; 170 ; 190]
Amp_3 (m) : [0.25 ; 0.25 ; 0.15 ; 0.07 ; 0.04 ; 0.04]

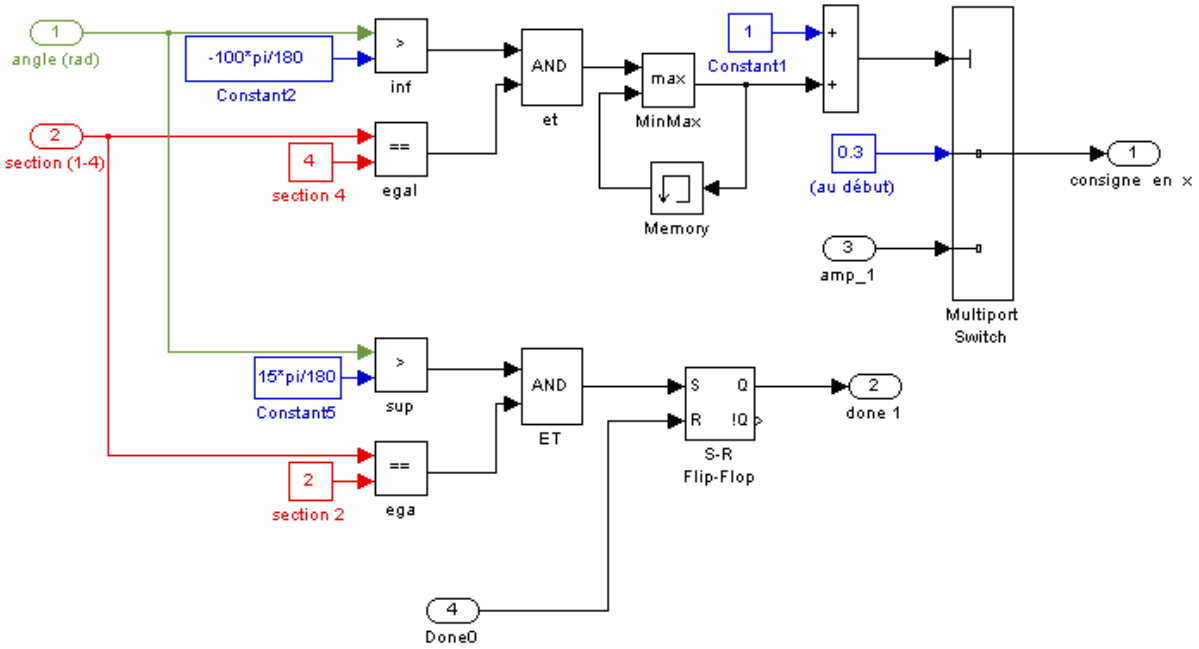


Figure A.V.7 : Contenu du bloc 'routine 1'

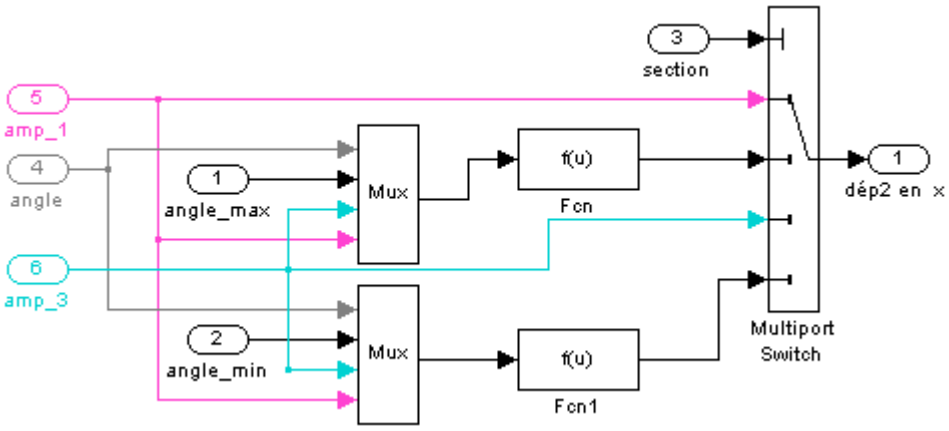


Figure A.V.8 : Contenu du bloc 'routine 2'

Fcn : f(u)= (-u(4)*u(1)/u(2)+u(3)*(1-u(1)/u(2)))*cos(u(1)*pi/u(2))

Fcn1 : f(u)=-(-u(4)*u(1)/u(2)+u(3)*(1-u(1)/u(2)))*cos(u(1)*pi/u(2))

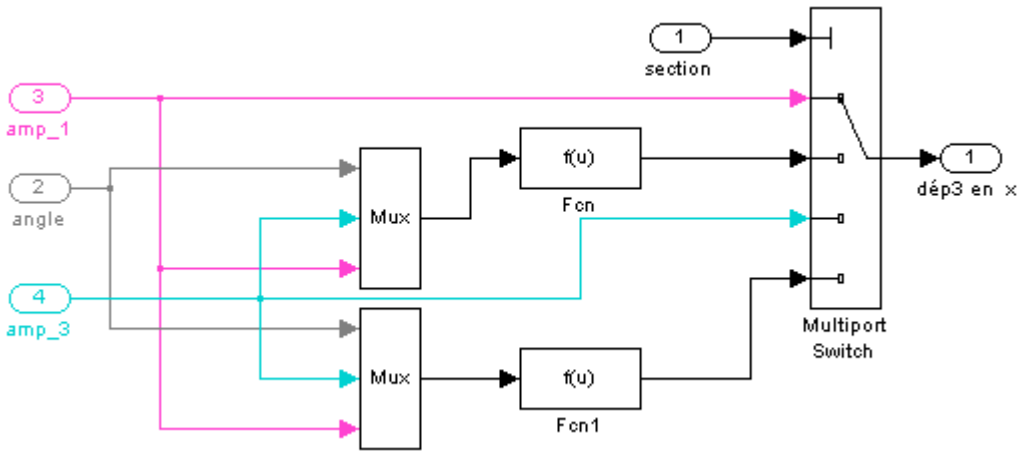


Figure A.V.9 : Contenu du bloc 'routine 3'

Avec :

Fcn : f(u)= (-u(3)*u(1)/100+u(2)*(1-u(1)/100))*cos(u(1)*pi/100)

Fcn1 : f(u)= (-u(3)*u(1)/100+u(2)*(1-u(1)/100))*cos(u(1)*pi/100)

ANNEXE VI : Schéma de câblage

Schéma de câblage entre le bloc de commande et la carte DSP